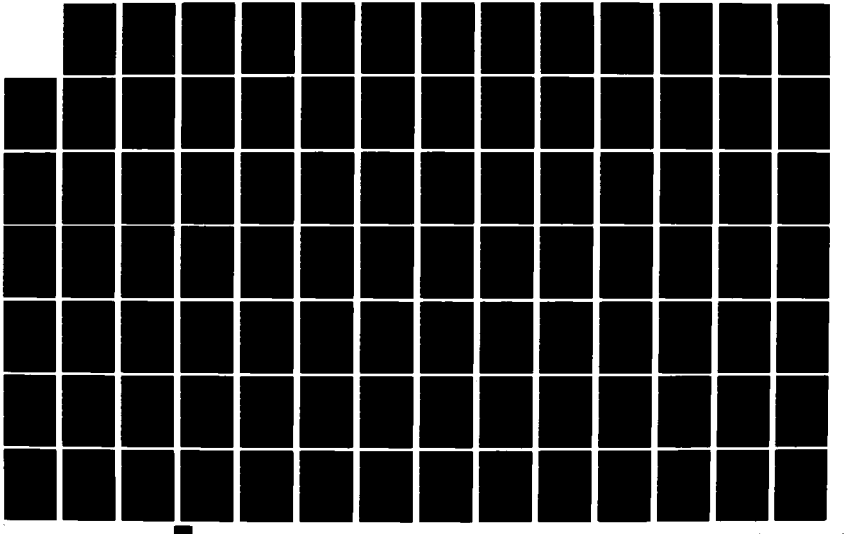
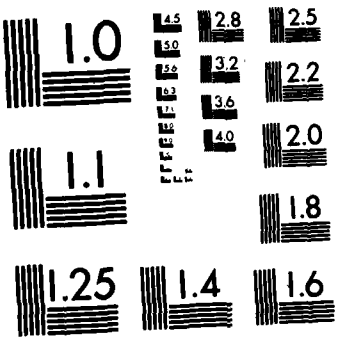
AD-A124 899 A STUDY OF MULTIVARIATE STATISTICAL ANALYSIS TECHNIQUES FOR COMPUTER PERF.. (U) AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH SCHOOL OF ENGI.. G MAGAVERO 1/2

UNCLASSIFIED DEC 82 AFIT/GCS/EE/82D-23 F/G 12/1 NL





A STUDY OF MULTIVARIATE STATISTICAL ANALYSIS TECHNIQUES FOR COMPUTER PERFORMANCE EVALUATION

THESIS

FIT/GCS/EE/82D-23 Gregory Magavero
Civilian USAF

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY (ATC)
AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

Unclassified
SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE	READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER AFIT/GCS/EE/82D-23	2. GOVT ACCESSION NO. AD-A124 899
3. RECIPIENT'S CATALOG NUMBER	
4. TITLE (and Subtitle) A STUDY OF MULTIVARIATE STATISTICAL ANALYSIS TECHNIQUES FOR COMPUTER PERFORMANCE EVALUATION	5. TYPE OF REPORT & PERIOD COVERED MS Thesis
	6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Gregory Magavero Civilian	8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Air Force Institute of Technology (AFIT-EN) Wright-Patterson AFB, Ohio 45433	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS	12. REPORT DATE December, 1982
	13. NUMBER OF PAGES 126
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)	15. SECURITY CLASS. (of this report) Unclassified
	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES Approved for public release; IAW AFR 190-17

4 JAN 1983

Approved for public release: IAW AFR 190-17.
LYNN E. WOLAVER
Dean for Research and Professional Development
Air Force Institute of Technology (ATC)
Wright-Patterson AFB OH

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)
Multivariate Statistical Techniques
Computer Performance Evaluation
Statistical Analysis

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)
This thesis was designed to provide a reference to computer performance analysts concerning the application of multivariate statistical techniques as alternatives to linear regression analysis in CPE. It provides some examples of the use and usefulness of Ridge Regression, Automatic Interaction Detection, Cluster Analysis, Canonical Correlation Analysis, Factor Analysis, and Discriminant Analysis.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

Unclassified
SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

AFIT/GCS/EE/82D-23

A STUDY OF MULTIVARIATE STATISTICAL ANALYSIS TECHNIQUES FOR COMPUTER PERFORMANCE EVALUATION

THESIS

AFIT/GCS/EE/82D-23 Gregory Magavero
Civilian USAF

Aprroved for public release; distribution unlimited

AFIT/GCS/EE/82D-23

A STUDY OF MULTIVARIATE STATISTICAL ANALYSIS TECHNIQUES FOR COMPUTER PERFORMANCE EVALUATION

THESIS

Presented to the Faculty of the School of Engineering
of the Air Force Institute of Technology
Air University
In Partial Fulfillment of the
Requirements for the Degree of
Master of Science

by
Gregory Magavero, B.A.
Civilian USAF
Graduate Computer Systems
December 1982

Approved for public release; distribution unlimited.

Acknowledgements

It is my hope that this thesis may provide a useful reference to computer performance evaluators by giving them some idea of the alternatives to linear regression analysis. I would be very pleased if this work was extended by others for the benefit of the CPE community.

For all their help in creating this thesis, I wish to thank my thesis committee. Dr. Thomas C. Hartrum served as my advisor; Lt. Col. James N. Bexfield and Maj. Joseph W. Coleman served as my readers. Their dedication and knowledge have greatly improved the quality of this thesis. I would like to acknowledge the great amount of assistance given to me by Dr. Charles W. McNichols in teaching me most of what I now know about the statistical techniques used and in providing two of the techniques in computer program form.

I would like to thank those people of the United States Air Force who made it possible for me to attend the Air Force Institute of Technology. I hope that I may repay the Air Force through the use of the knowledge gained there.

I would also like to thank my wife, Judy, and my family for putting up with me during the hectic months at AFIT.

Contents

Contents

List of Figures

List of Tables

Abstract

Computer performance analysts have relied too much on the use of linear regression analysis because of a lack of other data analysis techniques. Now, however, there are a number of easily available multivariate statistical techniques that need to be tested for applicability to Computer Performance Evaluation (CPE). This thesis presents some examples of how these techniques might be applied in this area, as well as some conclusions on their usefulness and applicability.

Ridge Regression appreared to be useful when the independent variables were multicollinear and an explanatory model was desired. Automatic Interaction Detection was helpful in detecting the structure of data, as was Cluster Analysis. Canonical Correlation Analysis was capable of reducing the dimensionality of data while relating two sets of variables, but was considered only somewhat applicable to CPE. Factor Analysis is similar in that it can also reduce the dimensionality of data by combining variables into common factors. Both techniques work best when the variables are multicollinear. Factor analysis was also judged to be of limited applicability to CPE. Discriminant Analysis seemed to generally be non-applicable to CPE because it was designed to work with nominally scaled data and CPE data tends to be intervally scaled.

Chapter 1 Introduction.

It is well known that one of the most investigated areas of computer technology is in seeking to improve computer system performance. As a formal study, this is known as computer performance evaluation. This thesis is concerned with one of the major approaches to computer performance evaluation, statistical analysis.

Computer performance evaluation data is characteristically multivariate in nature. That is, each observation consists of observed values for several variables, which may or may not be correlated. Statistical analysis of such data requires the use of multivariate statistical techniques.

While multiple linear regression analysis is the most widely used multivariate technique (in the sense that there is more than one variable per observation), there are two related problems in applying it to the analysis of computer performance data. First, multiple linear regression analysis requires certain assumptions of the population from which the data was selected. Among these assumptions are that no predictor variable can be formed by a linear combination of other predictor variables, and that the criterion variable can be formed from a linear combination of predictor vari-

ables plus an uncorrelated, normally distributed error term which has mean zero.

The second problem is that in many cases the results obtained with regression analysis are poor in the sense that the predictor variables explain only a small fraction of the variablility found in the dependent variable. The effect of poor results is that changes in the dependent variable cannot be explained in terms of changes in the predictor variables. Therefore, the value of new observations' dependent variables cannot be made with reasonable confidence based solely upon the (known) values of the predictor variables.

The assumption of a linear relationship between the predictor variables and the dependent variable may be relaxed if non-linear regression is used. The use of non-linear regression is difficult because the set of equations to be solved is also non-linear (ref 1). The least squares solutions are not generally obtainable in closed form. For this reason, iterative procedures must be used. The simplest method involves making a large number of data transformations and performing multiple linear regressions using the transformed data. This method has the drawback that there is no guarantee that any of the transformations will result in a good solution. More mathematically rigorous methods of solutions exist but are more difficult for the non-mathematician to use. As with the transformation method, these methods require the user to make some guesses about the

form of the equations and provide guesses or limits for coefficients. If the guesses are not good it may take a long time for the calculations to converge, or they may diverge.

Because of these problems, other multivariate statistical analysis procedures should be investigated for their applicability and usefulness in computer performance evaluation.

Scope

The statistical evaluation methods to be considered in this thesis will be limited to those existing multivariate analysis techniques which are available in program form on the CDC Cyber computers at Wright-Patterson Air Force Base.

These techniques will be tested using available Computer Performance Evaluation Simulation (CPESIM) data. CPESIM is discussed in Appendix A. The emphasis will be on experimentation based upon altering the configuration and workload of the simulated computer system.

The multivariate techniques to be investigated are Canonical Correlation Analysis, Discriminant Analysis, Factor Analysis, Cluster Analysis, Automatic Interaction Detection (AID), and Ridge Regression. The first three of these are available in SPSS (Statistical Package for the Social Sciences), the fourth is in BMDP (Biomedical Programs), and the last two exist as independent programs. AID was developed by the University of Michigan's Institute for

Social Research. The version used in this thesis originated at Lackland AFB and was adapted for use at the University of Texas, and modified by L. L. Gooch for use on a CDC 6600 computer. The use of this program is discussed in his unpublished Ph. D. dissertation for the University of Texas at Austin. The Ridge Regression program was written by Professor Charles W. McNichols at the Air Force Institute of Technology based upon Efroymson's Algorithm appearing in "Mathematical Methods for Digital Computers".

Assumptions

It is assumed that the CPESIM data being used in this project provides a fair representation of the complexity of real computer systems so that any conclusions concerning the applicability or usefulness of the various statistical techniques may be generalizeable.

Further it is assumed that the predictor variables available from CPESIM are neither greatly inferior nor superior to those commonly available from real computer system data. The predictor variables used are:

cpu	the amount of cpu time used by a job (seconds)
memory	the amount of central memory used (k-bytes)
cards	the number of cards read in
lines	the number of lines of output printed
diskio	the number of disk accesses made
tapeio	the number of tape accesses made
tapes	the number of tapes to be mounted concurrently
tarriv	the time when the first card of a particular job is read in (expressed in decimal hours)
iotime	total I/O time (dependent upon diskio, tapeio)

The criterion variable is:

turn job turnaround time (decimal hours)

Current Knowledge

Canonical Correlation Analysis attempts to find an optimal linear relationship between two groups of variables. There is no requirement that one set of variables be dependent upon the other set. However, within sets, the variables should be logically related. No previous application of this technique to Computer Performance Evaluation (CPE) was found.

Discriminant Analysis involves grouping data observations based upon some known criterion (criteria); finding a discriminant function of the observed values which maximizes the distance between group centroids; then placing observations, with known true group affiliation, into groups based upon this function. The success of correctly predicting group membership gives a measure of the real difference between groups. No previous application of this technique to CPE was found.

Factor Analysis is a technique used in an attempt to discover underlying factors in the independent variables that capture a large portion of the cause of data variability while reducing the dimensionality of the model to a more manageable level. No previous application of this technique to CPE was found.

Cluster Analysis techniques vary, but the main purpose

is to group observed data or variables based upon minimum distance (figured in any number of ways) between observations and known groups, or simply between ungrouped observations. T.C. Hartrum and J.W. Thompson (ref 9) have applied cluster analysis to computer performance modelling based upon jobs clustered by workload statistics. A.K. Agrawala (ref 2) and various associates have applied cluster analysis to grouping jobs by workload characteristics, and providing benchmark workloads. The results obtained were promising but pointed out the need for further investigation.

Automatic Interaction Detection (AID) is similar to cluster analysis. It is intended to assist a researcher in determining what, if any, interaction or quadratic terms need to be added to a regression model. This technique has been previously applied to determine its usefullness in CPE by M. Stover (ref 15). Her results do not appear to be generalizeable but she stated that AID provided insights into the nature of the data that would not have otherwise been discovered.

Ridge Regression is a variation of ordinary least squares regression analysis that attempts to correct for multicollinear relationships between predictor variables in order to more accurately predict values for the criterion or dependent variable. This method was also applied to CPE by Stover. Its potential usefulness could not be evaluated because the data used exhibited virtually no multicollinear-

ity (ref 15).

Approach

The approach taken in entering into this project was to first learn how to apply the techniques in general. Second, the techniques were investigated for theoretical applicability and for prior application to computer performance evaluation. Third, the techniques were applied to the analysis of data obtained from CPESIM based upon varying levels of complexity of the simulation parameters. They were observed for their explanatory as well as predictive abilities at successive levels of complexity in the simulation, and therefore in the data produced. Each method was compared, if applicable, to multiple linear regression analysis, which was used as a baseline performance standard, to determine its relative abilites. This third part was the major portion of the thesis effort.

Materials and Equipment

All the techniques investigated are available on the Cyber computers at WPAFB, either as part of SPSS, BMDP or as separate programs.

The simulated data used in the thesis was obtained from the CPESIM program which runs on the Cyber.

Expected Results

It was expected that this project would determine the

applicability and usefulness of various multivariate statistical techniques in providing beneficial analysis of computer performance data.

Chapter 2. Regression Analysis

Overview

Multiple linear regression analysis is being used as a baseline procedure in order to determine how well computer performance data may be characterised and how well predictions of future performance may be made based upon current data. In the succeeding chapters other, less well known statistical techniques will be applied to the same data. While the applications will try to be consistent with the theoretical requirements and abilities of the techniques, the results produced by these techniques will be compared to those produced by multiple linear regression analysis where applicable. The peculiar properties and abilities of each of the techniques will also be discussed.

Theoretical Considerations

The properties and use of regression analysis are well explained in many references, such as Introduction to Statistical Analysis by Dixon and Massey (McGraw Hill, 1951), therefore this section will be brief.

Regression analysis is used to obtain information about an environment based upon past data produced by that environment. The information obtained is in the form of estimated

regression coefficients which indicate the relative importance of one or more independent predictor variables in determining the value of a single dependent variable. This information may be used to predict new values of the dependent variable based upon known values of the independent variables or it may be used to explain the relationship between the dependent variable and each of the independent variables.

A measure of the predictive or explanatory power of the information obtained through regression analysis is the proportion of squared variation in the dependent variable that is explained by variation of the independent variables. This is called the R-squared value or the coefficient of determination (ref 10).

While regression analysis may use either a linear or a non-linear model, as mentioned earlier, the solution of non-linear models and the interpretation of their results is much more complex. For this reason, linear regession analysis is more commonly used. The model created can be expressed in the form of the following equation:

$$y_i = b_0 + \sum_{j=1}^{n} b_j x_{ij} + \epsilon$$

where:

y_i is an actual observed value of the dependent variable
b_0 is a constant or intercept term
each b_i is the calculated coefficient for one of the independent variables x_{ij}
ϵ is an error term equal to the difference between the actual and calculated values of y.

In matrix notation $\underline{b} = (X'X)^{-1} X'\underline{y}$

where:

$\underline{b}$ is the array of coefficients to be calculated and
X is the data matrix.

The assumptions required for linear regression analysis are that the error term ϵ is normally distributed with mean zero, the error terms for all observations are independent of each other and are identically distributed, and they are independent of the value of the dependent variable.

Objective of the Analysis

The objective of this part of the thesis was to create an explanatory/predictive model of simulated computer job turnaround time in terms of the independent job parameters produced by the simulation. It is desirable that the model explain a large percentage of the variation of the dependent variable (turnaround time in this case) in order that accurate predictions of future job turnaround times can be made. Using these predictions, the performance of a computer system executing a job with these particular characteristics could be estimated. Prediction of job performance also provides a means for model evaluation. Being able to explain a large percentage of variation of the dependent variable is also important because it indicates that the independent variables adequately characterise the dependent variable. The explanatory ability of the model helps determine possible computer system bottlenecks that hamper system performance.

Prior Applications

H. Gomaa (ref 7) has performed a regression analysis similar to the one used in this thesis as a baseline. He modelled the batch workload of a CDC 6000 Kronos system in the presence of a time sharing workload. Two types of independent variable were identified for use in his model. The first type measured job resource demands, and the second type measured the computer system workload while the job being observed was in execution. This latter type of variable is undoubtedly very important in explaining job performance, but is generally available only in gross terms at best. The basic model created provided estimates for elapsed job run time based upon values of CPU time, the number of concurrently executing short jobs, and the number of job steps in the routine being run. An improved model also took into account the changes that occurred in the number of concurrently executing jobs over the time period of the execution of the job being observed.

Gomaa's model included only short jobs, which had priority over long jobs for obtaining central memory. In addition, an analysis of residuals revealed that during periods of heavy workload some jobs which had a long turnaround time also had large positive residuals. Investigation revealed that these were actually short jobs that had been rolled out of memory for long periods of time. Because roll-out time was not an available job parameter,

Gommaa decided to exclude these jobs. "When the jobs with large residuals were excluded, models with much better fits were constructed," was Gomaa's (unsurprising) conclusion.

Later in the study period, a new computer was installed to run batch jobs exclusively. Removal of the interactive workload that had been present on the other computer resulted in an improved regression model (an R-squared value of .67 was mentioned). Still, about 5% of the jobs were highly I/O bound and therefore distorted the results. These were also excluded.

Finally, five models were constructed. Four represented data collected at different times over the four month period of study, and the fifth was cummulative. The models were "validated" by comparing the residual sums of squares of the individual models to that of the cummulative model. The regression coefficients were quite stable over the models, although the constant term varied. The results of the model which took into account the changes in the number of concurrently running short jobs produced slightly better results.

While Gomaa's results, in terms of R-squared values (the .67 figure mentioned) were better than those achieved in the analysis of complex data for this thesis or the work done by Hartrum (ref 10), he had two advantages not generally enjoyed. First, it is unusual to have a measure of the number of jobs running concurrently, and second, he seemed to

remove many jobs that did not fit well into his model. On the other hand, Gomaa used only three independent variables, while ten were available from the simulation data used in this thesis. Apparently his were better predictors of run time for those jobs not removed from the data set.

Another study with similarities to that used in this thesis was performed by Madhav Marathe (ref 11). This study compared system performance based upon different system configurations involving different central memory sizes and different types of disk drives. This is similar to the type of experimental design used in this thesis, which involved changing levels of demand for and availability of system resources in the simulated computer system and thereby varying the complexity of the environment in which the data was created. observed were that Marathe measured performance by the execution time of several command sets while this thesis made use of turnaround times of simulated jobs. Also Marathe was performing actual evaluation of computer performance while this is a study of the abilities of statistical analysis methods.

Approach

Computer job turnaround time was modelled using the available (independent) predictor variables. Three data sets were used for the initial ordinary least squares (OLS) multiple linear regression analysis. These data sets were

created with the Computer Performance Evaluation Simulation (CPESIM) program based upon varying degrees of complexity in the underlying environment.

Data set 1 consists of 787 observations representing computer jobs submitted by four different organizations, each with its own characteristics, over a five day (8 hours per day) period. These jobs exhibited a fairly high degree of competition for computer resources such as tape drives and central memory space. Data set 1' was created from data set 1 by using I/O time, a dependent variable, as a predictor of turnaround time.

Data set 2 consists of 106 jobs submitted by one organization over a four day period. There is very little contention for computer resources. This data set was also extended to 452 jobs for use with Automatic Interaction Detection, which requires large data sets. The extension was accomplished by increasing the simulated collection period from four to twenty days. This extended data set will be referred to as "data set 2+." This data is the simplest. Virtually no competition for computer resources exists in it because of the nature and timing of the submitted jobs.

Data set 3 consists of 90 jobs submitted by one organization within a one day period. These jobs exhibit a small amount of contention for computer resources.

Because OLS regression analysis was used only as a baseline against which to compare the other techniques, these

data sets were analysed using OLS in a stepwise fashion and the results saved for comparison to the results from other techniques. No data transformations were made in attempts to discover non-linearities or non-additivity.

The actual calculated regression coefficients will not be stressed because the emphasis of this thesis is on the predictive and/or explanatory power of the techniques rather than the actual results of an analysis. The regression equations obtained are shown in Table 2.1.

Because the regression technique being used adds variables in a stepwise fashion, the most significant predictor variable, in terms of the amount of reduction in the error sum of squares obtained, is added first and the procedure is repeated with the remaining variables until one of several stopping criteria is met. The stopping criteria may be specified by the user and include the maximum number of variables to be entered into the model and the minimum significance (F statistic) acceptable for entrance. The user may later decide to eliminate all variables entered after some point based upon significance values or the increment achieved in the R-squared value. These would be eliminated because the user felt that they did not add enough value to the model to offset the increased complexity they represent.

In work done for this thesis, the rule for limiting the number of variables accepted is to accept variables up to and including the first that results in an increment to R-squared

Table 2.1

Regression Equations for Data Sets 1 - 3

Data Set 1

$$turn = .00555\ mem + .160\ tapes + .0943\ tarriv + .000103\ diskio + .000492\ cpu + .000228\ tapeio - .943$$

Data Set 2

$$turn = .0000205\ diskio + .0000227\ lines + .000277\ tapeio + .000295\ cpu + .0000165\ cards + .611$$

Data Set 3

$$turn = .344\ tarriv + .00570\ mem + .000685\ cpu + .000101\ lines + .000307\ cards$$

The variable names are explained as follows:

turn	the job turnaround time (dependent variable)
cpu	the amount of cpu time used by a job
memory	the amount of central memory used by a job
cards	the number of cards read in
lines	the number of lines of output printed
diskio	the number of disk accesses made
tapeio	the number of tape accesses made
tapes	the number of tapes to be mounted concurrently
tarriv	the time when the first (job) card is read in

of less than .01. This rule results in accepting a fairly large number of variables in some cases, therefore it is not proposed as a general rule for other research. It was chosen here because of the low R-squared obtainable in working with data set 1.

Results

Using data set 1, six variables were accepted. These produced an R-squared of only .424. These variables were (1) memory size, (2) number of tape drives, (3) arrival time, (4) number of disk accesses, (5) CPU time, and (6) number of tape accesses.

Using data set 2, five variables were accepted. They produced an R-squared of .947. These variables were (1) number of disk accesses, (2) number of lines printed, (3) number of tape access, (4) CPU time, and (5) number of cards read.

Using data set 3, five variables were accepted. They produced an R-squared of .667. These variables were (1) arrival time, (2) memory size, (3) CPU time, (4) lines printed, and (5) cards read.

It is readily apparent from the R-squared figures that the amount of contention for computer resources in the environments that produced each of the data sets had a great impact upon the predictive power of the regression model obtainable. The explanatory ability of the model helps

determine possible computer system bottlenecks that hamper system performance.

Chapter 3. Ridge Regression.

Theoretical Development

When a significant amount of multicollinearity exists among the independent variables, ordinary least squares (OLS) regression analysis produces coefficients that tend to be too large (in absolute value) or of the wrong sign. Furthermore the coefficients may be unstable, that is, a small change in the data may result in appreciable changes in the calculated coefficients (refs 12,15).

Ridge regression is a variation of OLS regression which attempts to improve the accuracy of the calculated regression coefficients when using data which exhibits multicollinearity. This method injects a small amount of bias into the calculation of the coefficients with the hope that a resulting decrease in variance will produce a more accurate set of coefficients than could be obtained with OLS.

The ridge regression algorithm solves for a vector of coefficients by solving the following equation shown in matrix algebra notation:

$$\underline{b}^* = (X'X + kI)^{-1} X'\underline{y}$$

X is the standardized data matrix
X'X is the sample correlation matrix for the p independent variables
k is the bias value, which is incremented in discrete

steps over a succession of iterations to produce estimated coefficients with varying amounts of bias injected into the equation

I is the p x p identity matrix

X'$\underline{y}$ is the vector of sample correlations between the independent variables and the dependent variable.

With k=0 this equation produces the OLS estimates for the regression coefficients.

The output of the ridge regression algorithm used (specified in Chapter 1) included normalized and unnormalized estimated coefficients, a graphical "ridge trace," and variance inflation factors (VIFs), all for each value of k. The ridge trace depicts graphically the stabilization or settling of the calculated coefficients as the value of the bias term k increases. VIFs are the diagonal elements of the inverse of the X'X matrix and are one measure of multicollinearity. They are calculated as:

$$VIF_i = 1/(1 - r^2) \quad \text{(with k=0)}$$

where r is the multiple correlation coefficient between the given independent variable and all other independent variables (ref 13).

These outputs are all useful in determining the need for ridge regression and the smallest amount of bias required to offset the multicollinearity. Multicollinearity is measured through the values of the VIFs and by instability of the regression coefficients as shown in the ridge trace (ref 15).

Through examination of the output, one or more "optimal"

values of k are selected. This selection may be based upon certain heuristics. The best known non-graphical heuristics are (1) all VIFs less than 10, and (2) all improper signs changed (ref 5, 6, 13). Those studies referenced have indicated that it is desirable to choose as small a value of k as possible, thereby minimizing the amount of bias introduced into the calculations. The reason for choosing the first of these heuristics is that orthogonal data would exhibit a VIF equal to 1. If the value of the VIFs exceeds 10 multicollinearity is suspected. Also the correction of coefficient signs is chosen because of the previously stated fact that the coefficient signs are sometimes computed wrong when performing OLS regression on multicollinear data.

Perhaps the best known, but least precise, method of selecting k is through examination of the ridge trace. The value of k is chosen such that the coefficient values are fairly stable (not changing rapidly). This heuristic was not used for this thesis because of its subjectivity.

Previous Applications to CPE

The only previous application of ridge regression to Computer Performance Evaluation that was found was the work done by Stover (ref 15). She modelled job turnaround time and I/O time using simulated data from the same CPESIM program used for this thesis. However, she only used one set of data, which did not exhibit multicollinearity. Because of

the lack of multicollinearity in her data, she chose to use the heuristic "all VIFs less than or equal to 1" to select the "optimal" value of k. Had she chosen either of the heuristics mentioned earlier they would have undoubtedly have indicated that a k value of zero should have been selected, that is, there was no need to introduce bias into the calculations because there was no multicollinearity in the data. This would have made her application useless. Rather than this, she chose to introduce a greater degree of bias (as required by her heuristic) than would seem appropriate given the established heuristics. To determine the relative effectiveness of this technique, using this heuristic, she selected 14 observations randomly from her data set and calculated relative errors between the calculated turnaround and I/O times and the actual times. She also chose the same records for calculating relative errors using the OLS estimates presented earlier in her study. Although the results are not directly comparable, because she used additional variables in the ridge regression model, the relative errors were substantially larger using the ridge estimates than using the OLS estimates. Therefore the technique/heuristic combination seemed to be inappropriate for the data set used.

Approach

This study approached the situation somewhat differently

than Stover did. Similar to the study done by McNichols (ref 13), results achieved with data exhibiting little or no multicollinearity were contrasted with results achieved with data having a moderate to high degree of multicollinearity.

The first data set used was the contrived data shown in Table 3.1a. The dependent variable y was regressed against three independent variables x1 - x3. The first two of these were orthogonal to each other while the third was nearly perfectly correlated with the second. This latter relationship resulted in an extremely high degree of multicollinearity (RL = 2458.9). A stepwise OLS regression was performed in which the variables were added in order, x1 to x3. This was done to show the effect of multicollinearity on the calculated regression coefficients. Ridge regression was performed using all three independent variables for the purpose of seeing how well it could compensate for the addition of x3 to the model in terms of correcting the calculated coefficients.

The second example was less contrived in that it used the data from data set 1. The amount of multicollinearity, as measured by the VIF, was altered from 1.2 to 8.3 by adding the variable "I/O time" and removing two insignificant variables. I/O time was highly correlated with the number of tape I/Os and the number of disk I/Os. Ridge regression was performed on both models to see how much bias each of the heuristics would require and to see how close the calculated

coefficients for the second model would come to those of the first, which had very little multicollinearity. The heuristics applied were (1) all VIFs less than 10, (2) all signs correct, and (3) all VIFs less than or equal to 1 (Stover's heuristic).

Based upon these results I hoped to be able to determine the value of using the ridge method of coefficient estimation for data exhibiting varying degrees of multicollinearity, and if the method proved useful, to determine the better heuristic for choosing k.

The third example, which used data set 2, differed from the others in that no highly correlated variable was added to the model. This example was added to show a parallel with the results achieved using factor analysis (Chapter 7) on the same, highly multicollinear data. This data set, with four predictors, had a multicollinearity index of 69.5. The regression coefficients calculated when including amounts of bias corresponding to OLS and the various heuristics were compard to see the degree of change resulting from the use of ridge regression.

Results

Table 3.1b shows the coefficients calculated by OLS for x1 and x2 both before and after the inclusion of x3. When x3 was included the coefficient of x2 changed dramatically, from 0.34 to 28.7. The coefficient of x3 was calculated to be

Table 3.1a

A Data Set with extreme multicollinearity

y	x1	x2	x3
1.0	1	1	1.1
2.0	1	2	2.1
3.0	1	3	3.0
4.0	1	4	4.0
5.0	1	5	5.0
6.0	2	1	1.0
7.0	2	2	2.0
8.0	2	3	3.0
9.0	2	4	4.0
10.0	2	5	5.0
11.0	3	1	1.0
12.0	3	2	2.0
13.0	3	3	3.0
14.0	3	4	4.0
15.0	3	5	5.0
16.0	4	1	1.0
17.0	4	2	2.0
18.0	4	3	3.0
19.0	4	4	4.0
19.5	4	5	5.0
21.0	5	1	1.0
22.0	5	2	2.0
16.0	5	3	3.0
24.0	5	4	4.0
9.0	5	5	5.0

Table 3.1b

Effects of multicollinearity and Ridge on coefficients

Method	Coefficients			Comments
	x1	x2	x3	
OLS	4.07	...	...	x1 alone
	4.07	0.34	...	x2 added to model
	3.84	28.68	-28.54	x3 added to model
Ridge k= 0	3.84	28.7	-28.5	same as OLS
k = .004	4.05	1.12	-0.79	all VIFs .lt. 10
k = .024	3.97	0.34	-0.01	all VIFS .le. 1
k = .026	3.97	0.33	0.00	all signs correct

-28.5. Because x2 and x3 are very similar, the great disparity in the calculated coefficients could be confusing. In truth, it indicates that OLS uses the difference between the values of x2 and x3 to "fine tune" the predicted value of the dependent variable. The ridge technique was fairly successful in correcting for the effect of multicollinearity when calculating the coefficient for x2. Using the heuristic "all VIFs less than 10" a k value of .004 was required. This resulted in a calculated coefficient for x2 equal to 1.12. "All signs correct" required k = .026 and resulted in a coefficient of 0.33. "All VIFs less than or equal to 1" required k = .024 and produced a coefficient equal to 0.34, the actual coefficient calculated before x3 was added to the model. The R-squared statistic did not suffer from the addition of the required bias. It varied only from .8030 with k=0 (OLS) to .7932 with k = .004 to .7922 with k = .026. In this case the technique was effective in aiding interpretation of the contribution of the variables and did not greatly reduce the predictive ability of the regression equation.

In the second example, two models with differing amounts of multicollinearity were formed by changing the variables used. In the first model, the multicollinearity index was only 1.2, indicating very little multicollinearity. All the VIFs were initially less than 10 and all signs were correct, so these two heuristics indicated that there was no need to

use ridge regression with this data. However, in order to satisfy Stover's heuristic, all VIFs less than or equal to 1, a very large amount of bias (k = .105) was required. It is impossible to determine if this much bias was actually required to produce calculated coefficients equal to those that would have been calculated if no multicollinearity existed within the data, but this seems to be a very large amount of bias to add to data showing very little multicollinearity.

In the model with the moderately low RL (8.3), the first heuristic (all VIFs less than 10) required a k value of 0.015, the second required k = 0.070, and Stover's required k = 0.110. The coefficients calculated with the varying amounts of bias are shown in Table 3.2. They are contrasted with the coefficients calculated for the same variables in the model with the low degree of multicollinearity. Surprisingly, again Stover's heuristic produced the coefficients (for the varibles which were multicollinear) that were closest to those calculated by OLS using the non-multicollinear data. They are, however, only slightly better for these variables than the "all signs correct " heuristic, which added one-third less bias. These coefficients are also slightly worse than those calculated using the first two heuristics for the non-affected variables.

Because the introduction of bias into the calculation of regression coefficients could affect the predictive accuracy

Table 3.2

OLS and Ridge regression coefficients (standardized)

Data Set 1 (I/O time not included, RL = 1.2)

k	Standardized Coefficients						
	CPU Time	Central Memory	Tape Drives	Lines Prntd	Disk I/Os	Tape I/Os	I/O Time
0.000	.123	.359	.223	.046	.156	.097	....
0.105	.118	.329	.211	.053	.145	.098	....

Data Set 1' (I/O time included, RL = 8.3)

k	Standardized Coefficients						
	CPU Time	Central Memory	Tape Drives	Lines Prntd	Disk I/Os	Tape I/Os	I/O Time
0.000	.112	.371	.235	.025	.053	-.132	.265
0.015	.111	.365	.231	.030	.084	-.059	.181
0.070	.109	.347	.222	.037	.105	.002	.117
0.110	.108	.335	.216	.040	.107	.015	.105

of the model, a sample of the data observations was taken and the estimated value of the dependent variable (turnaround time) was calculated. This was done only for the cases k = 0 to k = 0.070, corresponding to OLS and the first two heuristics.

Twenty-one observations were randomly chosen from among the population of 787. The relative errors for each model were calculated and compared. The results are shown in Table 3.3. The relative error was smallest in 9 (42%) cases using the OLS model, 2 cases (9.5%) with all VIFs less than 10, and in 10 cases (48%) using the model for all signs correct. Adding bias to the regression coefficient calculation results in decreased RL values. In this case the three values were very close, ranging from .4179 with no bias to .4163 with k = .070. It should also be noted that the relative error values were generally all contained in a range of 1% error and always within 8%. These results all indicate that there seems to be no real predictive difference between OLS and ridge as would be hoped. The fact that the ridge equation produced with the smaller amount of bias proved to be the best estimator in only two cases can be explained by the fact that it was an intermediate position between two nearly equally good predictors.

Table 3.4 shows the regression coefficients obtained for data set 2 using ridge regression with varying levels of bias corresponding to OLS and the three heuristics. It is plainly

Table 3.3

obs #	k value	Relative Error in Per Cent .000	.015	.070	best
26		-2.26	0.78	8.89	2
68		56.94	58.75	60.43	1
108		-39.34	-39.37	-39.71	1
157		-42.53	-42.81	-42.93	1
186		47.35	47.55	48.73	1
257		-43.66	-42.01	-38.13	3
313		-15.80	-15.78	-16.76	2
327		15.07	15.78	16.10	1
381		-57.54	-57.08	-56.27	3
468		-66.76	-66.60	-66.34	3
476		-56.57	-55.84	-55.28	3
500		185.99	184.32	178.83	3
537		11.86	11.62	10.75	3
635		-7.58	-8.42	-10.75	1
661		138.23	138.25	138.61	1
677		49.86	49.23	47.53	3
687		51.61	50.80	49.78	3
700		-12.99	-13.50	-14.77	1
713		-1.48	-2.84	-4.54	1
728		54.93	54.28	52.61	3
766		118.54	117.06	112.95	3
Average		5.11	18.29	18.08	
Absolute Average		51.28	51.08	50.99	

Table 3.4

Effect of Ridge Regression on Data Set 2 Coefficients

		Variable				
Heuristic	k-value	I/O time	Tape drives	Disk I/Os	Tape I/Os	R squared
OLS	.000	.00173	-.00094	-.00009	-.00150	.8334
VIFs lt 10	.012	.00051	-.00133	.00001	-.00025	.7553
Signs	.044	.00026	-.00124	.00000	.00000	.7199
VIFs le 1	.092	.00019	-.00102	.00001	.00006	.7093

evident that the use of ridge with increasing amounts of bias resulted in large changes in each of the calculated coefficients, as would be expected with this highly multicollinear data. This information allows us to make a parallel with the results obtained when applying factor analysis on this data. This will be discussed in Chapter 7.

Conclusions

The ridge regression technique did, in both examples, provide corrections for the distortions in the calculated regression coefficients caused by multicollinearity among the independent variables. In both cases, the heuristics "all signs correct" and "all VIFs less than or equal to 1" (Stover's heuristic) were approximately equally good, and were better than "all VIFs less than 10." A reasonable approach to aiding interpretation of the relative weights of the coefficients of several multicollinear variables would seem to entail looking at both of the first two heuristics mentioned above and choosing the one that requires the introduction of less bias into the model. The implication of this is that it is not at all inappropriate to consider Stover's heuristic.

While ridge regression is not generally concerned with aiding predictivity, it is hoped that it will not greatly impair predictivity. In the two examples shown, there was little if any negative impact in this regard.

Chapter 4. Automatic Interaction Detection

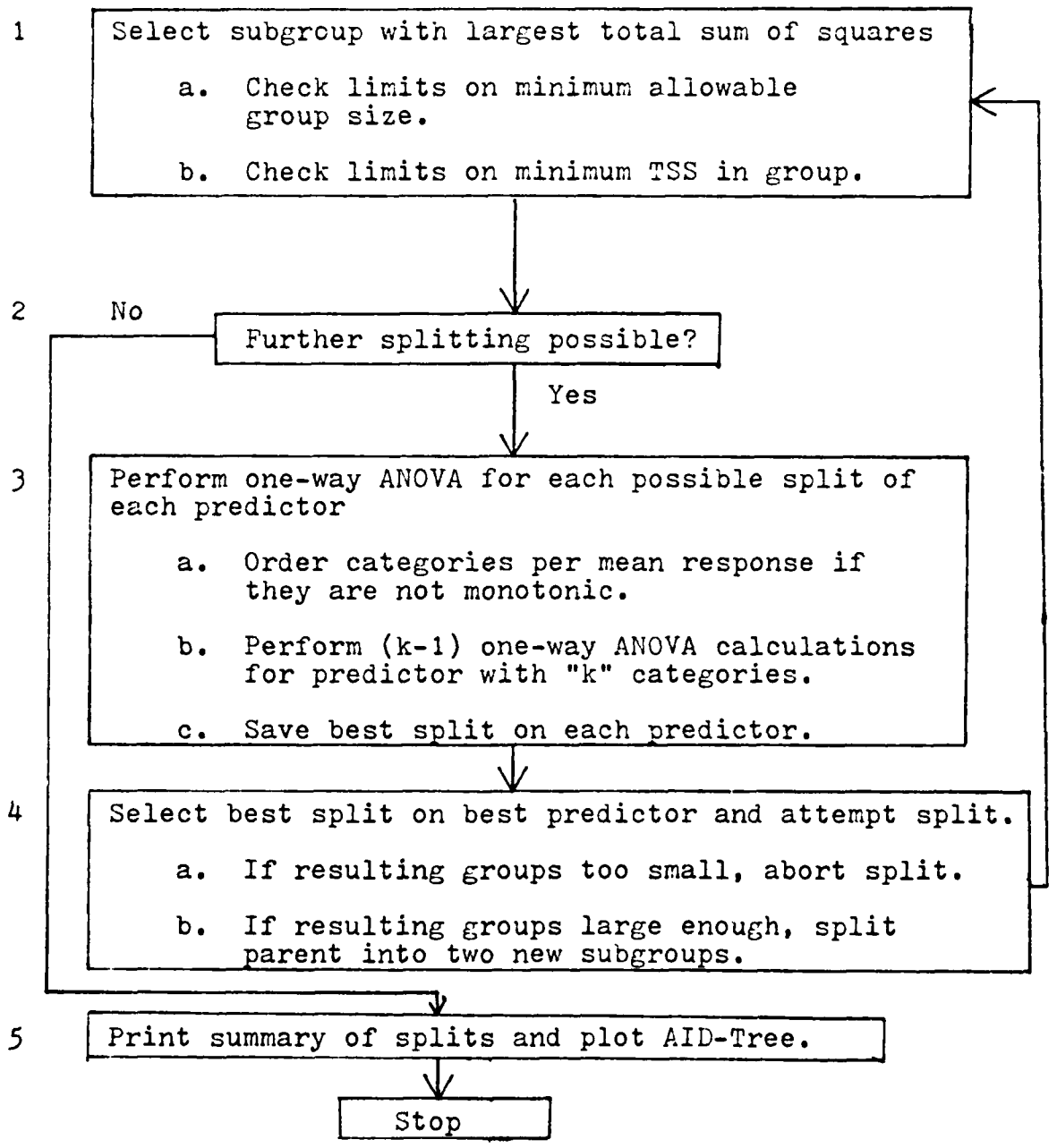
Theoretical Considerations

Automatic Interaction Detection (AID) was developed by J.A. Sonquist and J.N. Morgan at the Institute for Social Research, Ann Arbor, Michigan (refs 8,12). It is somewhat related to cluster analysis in that it partitions data into subset groups by using the split that will result in the greatest reduction in within groups variation. The major steps of the algorithm are shown in Figure 4.1 which is copied from Gooch's Figure 4.1 (ref 8).

The purpose of AID is to automate the process of searching for structure within raw data. According to the original hypothesis of Sonquist and Morgan, the proper model of the criterion variable using the given predictor variables is indicated by the variables used in the splitting process and the shape of the tree formed by the splitting. They hypothesized that an additive model would be shown as a symmetric tree. Therefore asymmetry would indicate interaction between the predictor variables, necessitating the addition of interaction terms to the (regression) model. Their research, using artificially created data, showed that an additive model would result in a symmetric tree, but later

Figure 4.1 (copied from Gooch)

Logic of AID Algorithm



Sonquist discovered that while additivity was sufficient, it was not a necessary condition for producing a symmetric tree. Signs of interaction (non-additivity) were a large increase in the predictive power of one of the variables after the effects of another variable had been removed by the splitting process, and asymmetry when the first split performed results in groupings with nearly the same mean value of the criterion variable.

Sonquist also determined that the order in which variables were used was not as reliable an indicator of their importance as was the total variation they explain. Gooch summarized Sonquist's recommendations for the proper use of AID. Three of these should be mentioned here because they may be violated in the work done for this thesis and possibly by others. He suggests using data sets of at least 1000 observations, having at least 40 observations per category (possible split), and discarding observations which are outliers on the criterion variable.

Requirements for use of the AID algorithm employed include specifying all values in integer form. This can be done by truncation, rounding, or by multiplying by a constant. The AID user must specify factor levels for each predictor variable at which splits may take place. These levels may be set at equally spaced intervals between the highest and lowest values of the predictor variables or they may be unequally spaced through explicit specification. The

existing data set at each level of interaction may be split only at the specified values. A one-way analysis of variance is performed for each possible split and the one which produces the largest reduction in within groups variation is chosen. This process continues until one of several stopping criteria, including minimum group size, maximum number of groups, and minimum amount of variation reduction, is met or exceeded.

After each split is performed a report is made about the split, including the increase in between groups sum of squares (BSS) and the increase in the ratio of BSS and the total sum of squares (TSS), and a t-value indicating the significance of the split. Also printed is a summary of all the groups existing after the split. Among the information printed is the total BSS for all groups identified, group sizes and means, and the R-squared value (total BSS/TSS).

When a stopping criterion is reached, in addition to a summary, two types of tree diagrams are produced. The first type is the skeleton tree which shows only the form of the tree produced by the splitting process. The second tree is detailed, giving characteristic values for each subgroup created. Only the first of these types of trees was available at the time that the work for this thesis was being performed. Figures 4.2 and 4.3 are augmented examples of the skeleton tree.

Because of the tree structure, the user is not required

Figure 4.2 Augmented AID Skeleton Tree

Data Set 2+

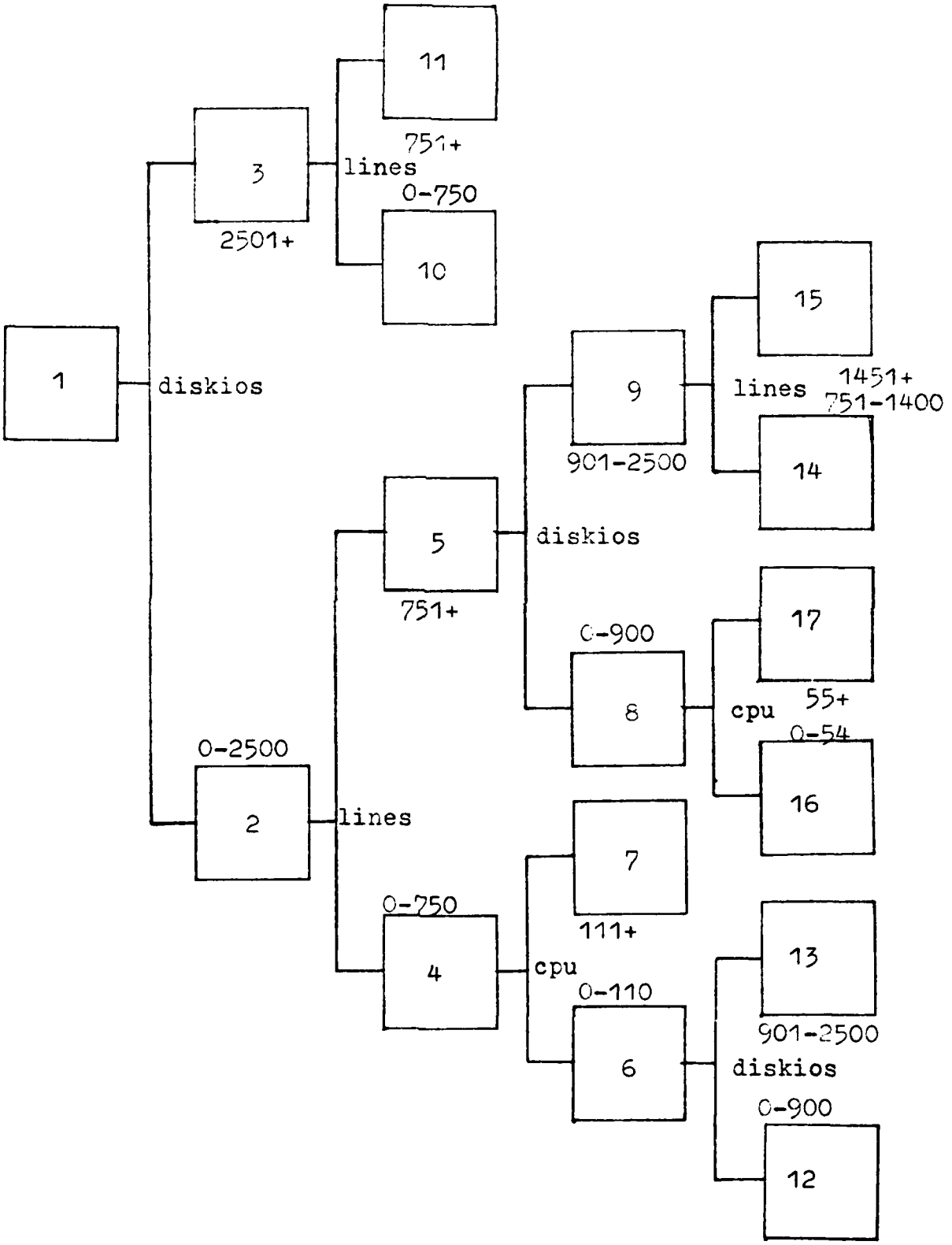
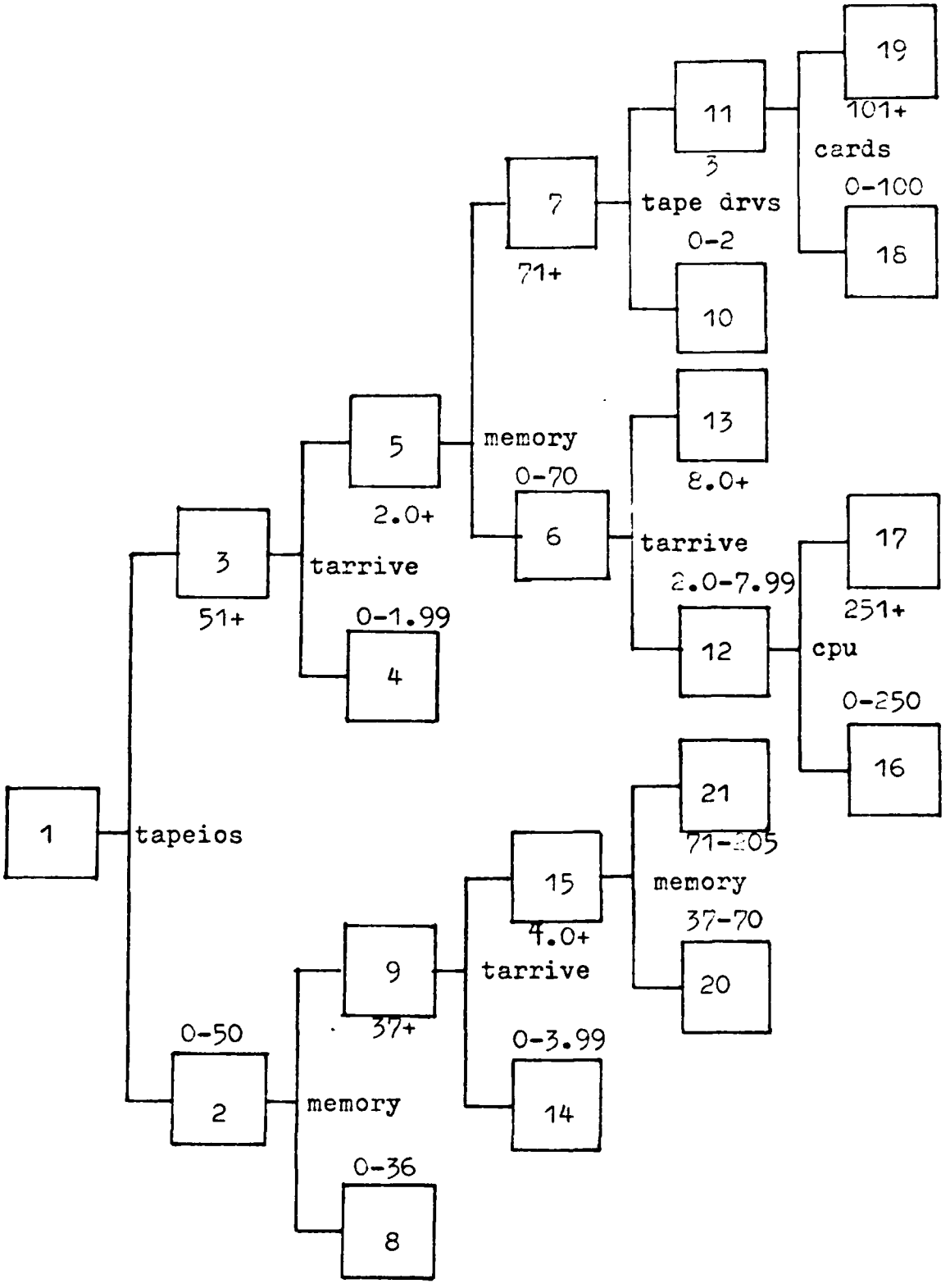


Figure 4.3 Augmented AID Skeleton Tree

Data Set 1



to accept the final result as produced by the algorithm. As with stepwise regression analysis, the process can be followed until additional splits result in insignificant gain in the R-squared value obtained. At this point all subgroups with higher group numbers (not just those on that particular branch) must be ignored. It is desirable to achieve a high R-squared value with a simple model, which is characterised by splitting performed on only a few different predictor variables. To quote McNichols (ref 12) "In AID an R-squared value close to 1 indicates that the partition process has sucessfully grouped observations with nearly identical values of the criterion variable, suggesting that knowledge of the predictor varibles values permits accurate prediction of the criterion variable value."

Previous Applications

This technique was applied previously by Margaret Stover (ref 15). In addition to a discussion of the requirements and capabilities of the AID algorithm, she provided detailed accounts of two AID analyses. In these analyses, she used CPESIM data with turnaround time as the criterion variable in the first analysis and I/O time as the criterion variable in the second. She discussed the relationship between the criterion variables and the varying levels of the predictor variables. She demonstrated the use of AID but did not provide generalizable information about applications of the

technique. She did not address the subject of interaction between predictor variables.

Experimental Design

To test the AID technique, two CPESIM produced data sets were used. These both modelled job turnaround time in terms of several predictor variables. The first of these (data set 1) was created from a complex simulated environment and as a result interaction was suspected among several of the predictor variables. The second data set (2+) was created from a simple simulated environment and no interaction was suspected. The purpose of this approach is to observe whether or not interaction is indicated by the shape of the AID tree and by the other indications previously mentioned. If interaction was indicated, an appropriate interaction term would be added to the regression model using a stepwise procedure to determine its usefulness. A second factor to examine in this technique is its ability to explain the structure of the data, which was partially known due to its creation via simulation.

Results

Modified versions of the AID trees are shown in Figures 4.2 (simpler data) and 4.3 (more complex data).

The AID tree for the simpler data set (2+) started out to be symmetric. The first split was by diskios (the number of disk accesses made) between level 4 (the most) and levels

0 - 3. The two resulting subgroups were both split by the number of lines printed between levels 0 - 2 and 3 - 4. At that point the splitting process in the upper half of the tree halted because one of the two subgroups was too small to be split again and splitting the other would not produce the minimum acceptable reduction in within groups sum of squares. That is to say that the elements of that group were too similar to allow further splitting. The bottom half of the tree, representing jobs with all but the highest level of disk I/Os (diskios) was further split through several levels. The splits after the second additional level (maximum of 4 splits to reach end node) were disregarded due to their minimal affect on the R-squared value. Although the variables by which the splitting took place in the bottom half of the tree were not used in the same order in each branch, they were same variables. These were the amount of cpu time, the number of disk accesses (again) and the number of lines printed (again). A total of nine end groups were identified. These are shown in Table 4.1. Included in the table are the group number, which identifies the order in which they were formed, the number of observations in each group, the mean value of the criterion variable (turnaround time) for observations in the group, and the levels of the predictor variables diskio, lines printed, and cpu time which characterize the groups.

Based upon the results observed and Sonquist's recommen-

Table 4.1

AID Clusters Formed with Data Set 2+

Group Number	Group Size	Mean Value of Criterion Var.	diskio levels	lines levels	cpu time levels
12	120	151.06	0 - 1	0 - 2	0 - 3
13	78	237.05	2 - 3	0 - 2	0 - 3
16	63	250.57	0 - 1	3 - 4	0 - 2
14	34	312.21	2 - 3	3	0 - 4
17	28	325.32	0 - 1	3 - 4	3 - 4
7	25	329.76	0 - 3	0 - 2	4
15	26	420.69	2 - 3	4	0 - 4
10	52	454.56	4	0 - 2	0 - 4
11	26	559.46	4	3 - 4	0 - 4

dation for determining whether or not a model is additive, it seems that AID agrees with the initial feeling that the data does represent an additive model. The tree is fairly symmetric and the variables in each branch of the tree are essentially the same. No increases in the explanatory power of any variable was observed when the effects of another predictor variable were removed by the splitting process. The technique seems to be sucessful showing that the model is additive.

In respect to showing the structure of the data set, AID also seems to have helped. It helped to identify the nine clusters shown in Table 4.1. Examination of their characteristics shows some very apparent differences between groups. For example, group 12 observations have a mean value for turnaround time of 151.06, the lowest of any group, and the observations are characterised by low levels for each of the three predictors. Group 11 shows the highest value for job turnaround time at 559.46 seconds and it has high levels of factors for the variables diskio and lines printed and was not split on cpu time.

The AID analysis of the more complex data set (1) had markedly different results. The tree was highly asymmetric. The first split was on the number of tape accesses, between levels 0-1 and 2-4. Subsequently, both resulting halves of the tree were split on arrival time and memory size, although not in the same order in all branches. At this point the near

symmetry ended. The number of cards read, the number of tape drives allocated, and the amount of CPU time used were used to split only one branch. This asymmetry is one sign of possible interaction between predictor variables. For this reason interaction terms representing the product of these three variables and the variables which preceeded them in splitting the branch on which they were located were added to the OLS regression model (with data set 1) discussed in Chapter 2.

Observation of the amount of variation (BSS/TSS) reduction possible by splitting on each of the variables at each step resulted in the detection of another sign of possible interaction. In the top half of the tree, the BSS/TSS for the number of tape drives allocated increased 31.9% when the split on arrival time was made, and increased an additional 46.9% when the split on memory size was made. Four interaction terms were added to the regression model to test for the significance of any interaction between these three variables.

Taking the first six terms (selected on the basis of the amount of increment in R-squared produced - see Chapter 2), the OLS model, without interaction terms, was able to achieve an R-squared value of only .424. With the addition of the described interaction terms an R-squared of .505 was achieved (taking the first seven terms). Included in these seven were the interaction terms representing the product of memory size

* arrival time * number of tape drives allocated (the most significant term), and the product of the number of tape drives * arrival time. No other interaction term was significant. The three variable product term was dropped and the regression re-run. The R-squared value achieved was again .505, however this time it was achieved with only six terms. The significant interaction terms were the product of memory size * arrival time (most significant) and the product of arrival time * the number of tape drives allocated (second most significant). The regression equation for this model was:

turn = -.425 + .00046 * cross1 + .0142 * cross3 + .00011 * diskio + .00050 * cpu + .0482 * tarriv + .00023 * tapeio.

where:

turn is job turnaround time,
cross1 is the product of memory used by time of arrival,
cross3 is the product of tape drives by time of arrival,
diskio is the number of disk accesses,
cpu is the amount of cpu time (seconds),
tarriv is the time of arrival (decimal hours),
tapeio is the number of tape accesses.

The interaction terms that proved significant were also intuitively appealing. Both the amount of central memory and the number of tape drives were limited resources for which there was a great deal of contention. As the day progressed, a job backlog was created and the contention for these resources increased. Therefore it is reasonable to believe that turnaround times would be longer for jobs arriving late in the day because of the waiting time required before

getting the named resources.

The net result of the analysis was that an apparent interaction was detected by the AID procedure, but that the most significant interaction, between memory size and arrival time, was somewhat surprising because following Sonquist's suggested procedure only allowed it to be found indirectly. Both these variables appeared in all branches of the tree and the BSS/TSS value of neither increased when a split was made on the other. Rather, the BSS/TSS of another variable, the number of tape drives allocated, increased when splits were made on both of these.

Conclusions

The AID technique did seem to be useful in detecting the existance or non-existance of interaction between predictor variables and in grouping the data into clusters based upon the relationship between the predictor variables and the criterion variable (as was shown for data set 2+). For the simple data set (2+) the lack of interaction was shown by the symmetry of the tree diagram. For the complex data set (1), the tree diagram was not symmetric and the amount of variation explainable by the number of tape drives allocated increased when the effects of the memory size and arrival time were removed by splitting on these variables. This led to finding variables which were interacting.

Chapter 5. Cluster Analysis

Theoretical Considerations

Cluster Analysis encompasses a large variety of methods for grouping data by either observations or variables. It is similar to AID except that it forms clusters based upon similarities in all the data, rather than similarities in the value of a criterion variable. It is not generally used by itself, but rather as a preliminary investigation into the structure of data. Recommendations for its use are a desire to place a collection of multivariate data into homogeneous groups, and that there is no explicit model from which to work (ref 12). Rather than having known group centroids, which would recommend discriminant analysis, the data itself must suggest grouping.

An important point about cluster analysis is elloquently stated by Anderberg (ref 3, p176).

> But the mechanical results derived from submitting a set of data to some cluster analysis algorithm are themselves devoid of any inherent validity or claim to truth; such results are always in need of interpretation and are subject to being discarded as spurious or irrelevant. The product of cluster analysis is not merely a set of clusters; the most useful outcome is increased understanding and improved organization of known facts permitting a more parsimonious description of the topic under discussion.

The results of a cluster analysis cannot be regarded in the same way as the results of a regression analysis because clustering data is not as straight-forward as fitting a line to a set of points using a least-squares algorithm.

Variation in Techniques

There are a great number of different useable clustering algoritms. The two points on which they all are based are that they must define a measure of similarity betwen objects and that they must define an algorithm for assigning the the objects to clusters. They may also address the question of how many clusters there should be.

Similarity may be measured by distance between the objects or by the "shape" of the objects. Euclidean distance is often used as a distance measure. The product-moment correlation is most often used as a measure of "shape." If the ratios of values of variables, for two objects, is nearly the same over all variables, then the objects have the same "shape."

Clustering may be performed on variables or on observations. Cluster algorithms are split into those that are hierarchical and those that are non-hierarchical. In non-hierarchical types, objects are assigned to clusters initially by either purposive or random assignment and through an iterative process. At each step they may be reassigned to another cluster into which they better fit.

This iterative process continues until an iteration is performed in which a minimal number (which may be user defined) of reassignments take place.

Hierarchical clustering does not allow reassignment, but rather starts either with all the objects in one cluster, or with each object in its own cluster of size 1. The latter is more common. In this case, the nearest clusters are iteratively combined until all the observations are in a single cluster. Those techniques that start with all observations in a single cluster, iteratively split the cluster into the sub-clusters which are farthest apart. This is repeated until each object is in its own individual cluster.

Both hierarchical and non-hierarchical clustering techniques have sub-variations, differing by such things as how distance between clusters is figured. These sub-variations are presented in Anderberg's book (ref 3), which is a very extensive and pleasing treatise on clustering.

Both types of clustering have advantages. Non-hierarchical clustering results in a distinct number of clusters as an end poduct whereas the user must decide where to cut off hierarchical clustering and determine what clusters exist at that point. This point is especially relevant because no standard procedure for making this decision was seen in the references cited. Hierachical clustering is computationally simpler and faster, and no initial estimates of cluster

positions are required. Also outliers may be more clearly shown than with some of the non-hierarchical techniques which force all objects into the nearest cluster. Not all non-hierachical techniques do this, however.

Previous Applications

Previously, cluster analysis has been used by A.K. Agrawala and J.M. Mohr and their various associates to characterize computer system workload. One article by them (ref 2) will be discussed below. T.C. Hartrum and J.W. Thompson jointly produced a paper (ref 9) discussing their use of cluster analysis to show the effect of a job scheduling policy change on computer performance. This is more closely related to the purpose of this paper, but unfortunately seems to be the only reported study of its kind.

In the report cited, Agrawala and Mohr used a non-hierarchical clustering algorithm to find workload characterising clusters. The computer accounting package used provided them with sixty-four factors. From these they chose three "feature sets" composed of four to six reported factors. The results shown were the number, size, and characterising values for clusters formed using each of the feature sets. A comparison of the feature sets was performed by constructing a "confusion matrix" which showed the number of jobs from each cluster produced by one feature set, that was placed in

a different cluster by a second feature set. If the feature sets are unrelated, this should show a random pattern. They indeed did conclude that the feature sets were unrelated and therefore represented different groupings of the population. They used this result to warn that care should be taken in choosing the feature set used to cluster data since the final clusters produced by one feature set may be entirely different from those produced by another feature set.

Later in the paper, the authors discussed ways to tell if the clusters formed are naturally occurring. They suggest as a first step to have large enough samples "to avoid dimensionality and sample size problems" (ref 2, p27). Next they showed how to demonstrate the correctness of the assumption they made that each cluster is distributed unimodally. This was done by plotting the distance from the cluster centroid to each of the observations in the cluster. Unimodality is characterised by monotonic decreases in the incidence of points falling within arbitrary distance categories. Not mentioned was the fact that unimodality, demonstrated in this way, could be "doctored" by changing the boundaries on the distance categories.

Another sign of natural groupings is that observations are ultimately placed into the same final cluster regardless of how the initial clusters were defined. This would be true for the non-hierarchical clustering method used by them, but would not apply to hierarchical methods. Finally, they

mention that an additional check for natural groupings is if they seem to make sense.

T.C. Hartrum and J.W. Thompson performed a cluster analysis that was more closely related to computer performance evaluation. Their idea was similar to that of Agrawala and Mohr, in that they wished to simplify the relationship between predictor and criterion variables by defining the relationship within identified clusters and ignoring the area outside of the clusters.

They clustered computer jobs based upon values of CPU time, I/O time, central memory size, and number of disk sectors used. This was done both before and after the implementation of a job scheduling policy change. This policy change involved adding a second computer to run most of the interactive jobs while keeping all the batch jobs on the original computer. This was done in an attempt to improve turnaround times for batch jobs.

Using a non-hierarchical clustering algorithm, Hartrum and Thompson identified a small number of clusters in the accounting data from both periods. By comparison of the mean values of each of the variables, they declared that four of the clusters had remained stable enough to conclude that they were present in both the before and after change data sets. They observed that each of these four clusters showed an average improvement in turnaround time. This varied from 35% to 49%, and was significant at the .001 level of signifi-

cance. Therefore they were able to conclude that, for the jobs in the identifiable clusters, the policy change was beneficial.

Approach

There are a number of reasonable uses for cluster analysis in the analysis of computer performance data. Aside from the uses mentioned above, cluster analysis could be used to test for stationarity of workload over some time period or to determine if computer jobs are clustered according to submitting organization. This latter could be useful in determining if the algorithm that computes computer time charges is fair to all organizations.

The example that will be shown in this chapter is another example of the type of analysis done by Hartrum and Thompson. This is a comparison of mean turnaround times for jobs within each identified cluster between data obtained before and after the computer system cpu time quantum was changed from 1 second to 10 seconds. Because the computer system was simulated the same workload could be submitted both before and after the change, thus maintaining the same clusters since only the job turnaround times changed. An overall comparison of the turnaround times of all jobs could have been tested to determine the effect of the time quantum change by comparing the overall mean times or by performing a multiple linear regression analysis, but it was hoped that

additional detail would be available if the analysis was performed with cluster analysis.

The BMDP clustering algorithm used (P2M) for this thesis standardized all variable scores before determining the distance between observations. Therefore the distance between clusters was the standardized distance. This distance was also calculated between the (size) weighted group centroids. All observations were weighted equally. Alternatives to the procedure used are available in BMDP. These include non-equal case weights, alternative linkage methods, and the use of euclidean distance measures. There were also other options which could be used if the data consisted of frequency counts.

Results

The test showing the effect on job turnaround time of changing the cpu time quantum from 1 to 10 seconds yielded eight clusters. These are shown in Table 5.1. Please note that a large number (47 of 169) of observations are not included because they were in clusters of size smaller than 3 at the point at which the clustering was cut off. As can be easily seen in the table, neither time quantum produced better turnaround times for all jobs. The 10 second time quantum was better more often (82 of the 122 observations). The logical next step would be to determine which factors resulted in the increased turnaround times and which factors

Table 5.1

Characteristic values of clusters before and after change

Time Quantum		1 second		10 seconds	
Cluster Number	size	mean(y)	s.d.(y)	mean(y)	s.d.(y)
1	48	.460	.473	.394	.433
2	24	1.203	.079	1.184	.611
3	20	1.821	.540	1.828	.487
4	3	1.493	.351	1.689	.461
5	7	2.743	1.632	2.082	.908
6	3	.981	.143	1.196	.288
7	3	1.715	.466	2.148	.306
8	3	1.771	.440	1.859	.372

Standardized variable means (cluster centroids)

Cluster Number	Arrival Time	CPU Time	Central Memory	Tape Drvs	Cards Read	Lines Prntd	Disk I/Os	Tape I/Os
1	.139	-.658	-.761	-1.058	-.392	-.594	-.518	-.602
2	-.722	-.306	-.259	.398	-.355	-.239	-.127	-.224
3	1.096	-.146	.129	.827	-.287	.133	-.511	-.184
4	.460	2.023	-.164	.725	-.516	-.167	-.234	-.305
5	-.439	-.262	1.885	1.307	-.561	-.353	-.515	.106
6	-1.112	.876	1.384	-.148	1.498	.044	-.103	-.079
7	.842	.037	-.716	.652	-.287	.161	-.357	2.416
8	-.373	-.318	-.083	.434	1.554	.097	2.493	1.375

resulted in the decreased times. In an actual study of computer performance the trends in this data could be analyzed, but that is not the purpose here.

Conclusions

Clustering can be used profitably when the data set covers a wide dispersion which prevents meaningful analysis. By concentrating effort only upon data within well defined areas, more reasonable relationships can be attributed to the data. Also, clustering is recommended not as an independent statistical analysis procedure, but as a preliminary step in understanding the characteristics of a data set.

Chapter 6 Canonical Correlation Analysis

Theoretical Considerations

Canonical Correlation Analysis is used to relate factors which are described by more than one variable. An example of this might be to relate computer jobs size, as measured by I/O time and CPU time, to resource requirements, as measured by the number of job steps, number of I/O operations, and the amount of central memory required. This technique has been used extensively to measure attitude based upon answers to several different but related survey questions (ref 15). It is also used to measure intelligence by relating scores on several different types of tests to factors which demonstrate intelligence and deriving relative weightings to give to each of the test results. Normally when a small number of variables are to be analysed, multiple regression is used. However, when the number of variables increases beyond a point, meaningful interpretation of the results is impossible. It is necessary to somehow reduce the dimensionality of the data. Factor analysis is helpful in this case but is limited in that no distinction is possible between dependent and independent variables. This distinction can be made with canonical correlation simply by using the dependent variables

as one set and the independent variables as the other. It is immaterial which set is dependent and which is independent (ref 14). Because this technique relates two sets of several variables, it is considered to be more truely a multivariate statistical technique than multiple regression analysis, which has only a single dependent criterion variable.

Mathematically, this technique calculates canonical weights (similar to regression coefficients) for two simultaneous regression-like equations. Each of these equations shows the contribution of each variable to its associated factor (called the canonical variate). The goal of the technique is to find the set of weights that produces the greatest possible canonical correlation between the two canonical variates. A canonical correlation is the simple correlation between canonical variates. Standardized variable scores are used in this process in order to eliminate the need for constant terms in the individual equations. Using an example with two variables measuring job size and three measuring resource utilization the following equations would be simultaneously solved:

$$y^* = a_1 y_1 + a_2 y_2$$

$$x^* = b_1 x_1 + b_2 x_2 + b_3 x_3$$

subject to maximizing the absolute value of the canonical correlation, that is, the correlation between the canonical variates x^* and y^*. Although the use of the x and y notation for the variates is similar to the notation used in

regression analysis, there is no need for a dependency relationship of the y-variate on the x-variate. Rather, what is needed is that each of the variables combining to describe a canonical variate should be related to the others combined to describe that variate to the extent that they measure different, probably overlapping, aspects of that canonical variate.

Solution of canonical correlation analysis would result in an infinite number of solutions, but by applying constraints to the solutions, they can be limited to a small number. The actual number of solutions is equal to the lesser of the number of variables combining to describe either of the two canonical variates. These solutions will be unique only up to the point that they may be multiplied by any constant and still be proper solutions. The most advantageous use of this property is in changing all signs by multiplying by -1. Each solution consists of an eigenvector of canonical weights (a_i and b_i), and an eigenvalue which is equal to the square of the canonical correlation (the simple correlation between y* and x*). Normally the solution with the largest eigenvalue is accepted and the others are discarded because the largest eigenvalue is indicative of the best relationship between the canonical variates. An exception to this practice is in the case that several eigenvalues are nearly equal in magnitude. It may then be desirable to explore the alternative relationships offered.

One of the others may prove to be more intuitivly appealing.

McNichols (ref 12) shows the procedure developed by M.S. Bartlett (reported in Biometrica, January 1941) to perform a hypothesis test of the statistical significance of the probability that at least one of the eigenvalues is non-zero. This can be performed sequentially on all the eigenvalues by performing the hypothesis test, eliminating the most significant eigenvalue and repeating the test until a non-significant value is encountered. Wilk's Lambda is another test of the significance of the canonical correlation.

While interpretation of the canonical weights may give an idea of the relative importance to each of the variables, and may even suggest a name for the canonical variate, there is another, seemingly better, method of measuring the relative impact of each variable. This is through the use of canonical loadings. These are the correlation between the variables and their associated canonical variate. Those variables with high loadings are closely related to the canonical variate, and are therefore important for its formation.

Redundancy is a measure of the amount of variance in one set of variables (those used to describe one canonical variate) explained by the other set. The redundancy for each set of variables must be computed based upon the given condition of the other set. This is calculated for each set

of variables as the product of the average loading of the variables on their associated canonical variate times the squared canonical correlation, which represents the relationship between the canonical variates x* and y*. In equation form the redundancy for the x-variables (p of them) is:

$$(r^2_{y^*x^*})(\sum_{j=1}^{p} r^2_{x^*x_j})/p$$

Because average loading of the variables is used, the redundancy figure will actually decrease if an inappropriate variable is added to the model. The redundancy figure is analagous to the R-squared of multiple regression analysis because it measures the amount of variation in one canonical variate that is explained by the variation in the other canonical variate. Going one step further, the total redundancy over all solutions (all eigenvector/eigenvalue pairs) can be summed.

It should be noted that the redundancy of the x-variate given the y will generally not be equal to the redundancy of the y-variate given the x.

Previous Applications

No previous application of canonical correlation analysis to computer performance evaluation was found.

Approach

For the application of canonical correlation analysis the variables representing turnaround time and I/O time were used as a set of variables denoting job length. The

independent variables which measured resource utilization and arrival time were used as the second set. The reason for pairing the first set as such was that turnaround time and I/O time represent different, but related, measures of length. Unifying these in this way seems to be appropriate for the use of canonical correlation. This analysis was performed on data sets 1, 2+ (the long version of data set 2), and 3.

Results

Table 6.1 shows the results obtained using data set 1. Two solutions, or sets of canonical variates, were obtained. The information for each set of variates is presented in a two column format in the table. The first column of each pair contains the canonical coefficients of the variables and the second column shows the canonical loadings of the variables on the canonical variate. From top to bottom, the table shows the results for the x-variables followed by the redundancies for the x-variables given the y-variables. This information is repeated below for the y-variables. At the bottom of the table are the eigenvalues, canonical correlations, levels of significance, and degrees of freedom for the two solutions.

Several points should be mentioned about the results. First, the loadings of the dependent variables, turnaround time and I/O time, are high on different y-variates (y*).

Table 6.1

Results of Canonical Correlation Analysis of Data Set 1

Variate	CANVAR1 Canonical Coeff	CANVAR1 Canonical Loading	CANVAR2 Canonical Coeff	CANVAR2 Canonical Loading	Total
x-variables					
Arrival Time	.0070	-.0353	.4777	.4997	
CPU	.0105	.2998	.0845	.2573	
Memory size	.0068	.2334	.3789	.5366	
Disk I/Os	.2157	.3842	-.0695	.0415	
Tape I/Os	.9290	.9755	-.4147	-.0224	
Tape Drives	-.0019	.4612	.7743	.6981	
Cards	.0061	.2566	-.0179	.0350	
Lines	.0203	.2743	-.0662	.1526	
Redundancy		.1972		.0798	.2770
y-variables					
Turnaround time	.0026	.3713	1.0760	.9285	
I/O time	.9990	1.0000	-.3995	-.0024	
Redundancy		.5615		.2462	.8077
Canonical Correlation	.9935		.7558		
Eigenvalue	.9870		.5712		
Significance	.0001		.0001		
Degrees of Freedom	16		7		

I/O time is highly loaded on the first y-variate while turnaround time has a low loading on that variate. This situation is reversed on the other y-variate. This occurred because each of the dependent variables was related to a different set of independent variables. Therefore it would be more appropriate to solve for each of the dependent variables in a separate regression analysis.

Also note that the redundancy figures for the x-variables are quite low, showing that many of the independent variables are poorly related to both of the x-variates. Neither cards read nor lines printed loaded very highly with either x-variate so eliminating them should cause the redundancy of the x-variables to increase. Table 6.2 shows the results obtained after the elimination of these two variables. Notice that the redundancies did increase.

Analysis using data set 3 had the same problem with the dependent variables being related to different independent variable. For this reason the results will not be presented.

Data set 2+, which arose from the simplest simulated computer environment, did not exhibit this problem. Turnaround time and I/O time were dependent upon the same independent variables, probably because many confounding factors had been removed from the simulated environment. For example, jobs arrived at large enough time intervals so that they could be loaded into memory almost immediately upon arrival and begin execution shortly thereafter. Also there

Table 6.2

Results of Canonical Correlation Analysis of Data Set 1
(variables removed)

Variate	CANVAR1 Canonical Coeff	CANVAR1 Canonical Loading	CANVAR2 Canonical Coeff	CANVAR2 Canonical Loading	Total
x-variables					
Arrival time	.0072	.0344	.4867	.5008	
CPU time	.0146	.3003	.0722	.2570	
Memory size	.0097	.2343	.3724	.5370	
Disk I/Os	.2188	.3844	-.0801	.0405	
Tape I/Os	.9307	.9757	-.4251	-.0254	
Tape Drives	.0032	.4625	.7637	.6982	
Redundancy		.2400		.1038	.3438
y-variables					
Turnaround Time	.0051	.3734	1.0759	.9277	
I/O time	.9981	1.0000	-.4018	-.0047	
Redundancy		.5621		.2448	.8069
Canonical Correlation	.9933		.7542		
Eigenvalue	.9866		.5688		
Significance	.0001		.0001		
Degrees of freedom	12		5		

were very few concurrently executing processes.

The results obtained with this data set are shown in table 6.3. Turnaround time and I/O time were both highly loaded on the first canonical y-variate. The number of disk I/Os and tape I/Os loaded heavily on the corresponding x-variate, indicating that the two dependent variables were largely dependent upon the two measures of input-output.

As in the first example with data set 1, the loadings of the x-variables were quite low. They would increase in a similar manner if the variables arrival time and memory size were removed.

Conclusions

Canonical correlation analysis is an example of a statistical technique that should not be routinely applied to the analysis of computer performance data, but should be reserved for those cases in which a large number of variables are analyzed and sets of dependent and independent variables may be formed so that the variables within each set are reasonably related.

Table 6.3

Results of Canonical Correlation Analysis of Data Set 2+

Variate	CANVAR1 Canonical Coeff	CANVAR1 Canonical Loading	CANVAR2 Canonical Coeff	CANVAR2 Canonical Loading	Total
x-variables					
Arrival time	-.0004	.0126	-.0077	-.0949	
CPU time	.0040	-.0104	-.6379	-.6499	
Memory	-.0034	.0054	-.0350	-.0454	
Tape drives	-.0011	.2606	-.0158	.0054	
Cards read	.0112	.0788	-.3000	-.3323	
Lines printed	.0490	.0863	-.6908	-.6916	
Disk I/Os	.8702	.9183	.0474	.0154	
Tape I/Os	.3932	.4988	.0752	.0675	
Redundancy		.1453		.1057	.2510
y-variables					
Turnaround time	-.0369	.7969	-1.7040	-.6041	
I/O time	1.0297	.9998	1.3582	-.0217	
Redundancy		.8095		.1504	.9599
Canonical Correlation	.9952		.9072		
Eigenvalue	.9904		.8230		
Significance	.0001		.0001		
Degrees of freedom	16		7		

Chapter 7 Factor Analysis

Theoretical Considerations

Factor Analysis is used to search for a reduced dimensionality of data by finding "factors" or "latent variables" that convey approximately the same information as the manifestation (measured) variables. Use of this technique may also aid interpretation or naming of the factors. It is especially valuable when the manifestation variables exhibit multicollinearity.

There are many variations of factor analysis, and seemingly, much information discussing it. SPSS provides five variations (ref 14). These are (1) principal factoring without iteration (also known as principal component analysis), (2) principal factoring with iteration, (3) Rao's canonical factoring, (4) alpha factoring, and (5) image factoring. The differences between these methods are discussed in (technical) detail in the SPSS manual (ref 14) and by Bennett and Bowers (ref 5). They all have two things in common. First, all factors are orthogonal and second, factors are arranged in decreasing order of importance. It is also often true that the first factor tends to be general, having moderate loadings on most or all the manifestation

variables. This is because the first factor is the best one-dimensional representation of the data. Subsequent factors tend to be "bipolar," having high positive loadings on some factors and high negative loadings on some others. The first two techniques listed are the most well known and used.

The results of a factor analysis are reported as a number of eigenvalues, each representing one factor. The eigenvalues are equal to the squared loadings of the factors on the data. Initially there are as many factors identified as they are manifestation variables. The sum of the eigenvalues also equals this number, so the average value of all the eigenvalues is 1.0. Dividing each eigenvalue by the sum yields the percentage of total variation explained by that factor. Using the information available at this point, there are three common methods to decide upon the significance of the factors, enabling the user to eliminate subsequent factors and discover the true dimensionality of the data. First is Kaiser's criterion that only factors with eigenvalues of more than 1.0 should be retained. An eigenvalue of 1.0 corresponds to having the factor explain as much of the total variation as one original manifestation variable, or of being average in its explanatory power (since the average eigenvalue is 1.0). Bennett and Bowers proposed a criterion that each factor should explain at least ten per cent of the variance. This would result in the retention of

more factors than Kaiser's criterion if there are fewer than ten manifestation variables, but in the retention of less factors if there are more than ten manifestation variables. The third method is graphical and was proposed by R.B. Catell. It is called a "scree test." The eigenvalues are plotted on a two-dimensional graph with the vertical axis representing eigenvalue scores and the horizontal axis representing factor numbers. The plotted eigenvalues are then connected with straight lines. This will result in a graph which has several negatively sloped segments. Normally there will be an identifiable "elbow" in the graph where it becomes much flatter (the segments become more nearly horizontal). Solutions associated with each of the eigenvalues up to and including the one at the elbow should be retained. According to McNichols (ref 12) this test and Kaiser's criterion are normally within one factor of agreement. SPSS normally retains factors if the eigenvalue is at least 1.0.

Communality is a measure of variance that exists in common between factors and each variable. For each manifestation variable it is the sum of the squared loadings of that variable on all of the retained factors. It is a simple summation because all factors are constrained to be orthogonal and uncorrelated. Communality is the R-squared that would be obtained if the retained factors were used to predict the manifestation variables. For this reason it is a

measure of the sucess of modelling the data with the retained factors.

Factor rotation is usually performed on the retained factors to aid interpretation. Thurstone proposed a set of desirable traits for rotated factors. These are: (1) most loadings (of variables) on any factor should be small, with a few close to plus or minus one; (2) a specific row (variable) should contain non-zero loadings on only one factor; and (3) any pair of factors should exhibit different patterns of loadings. The more common rotational methods are attempts to achieve one or more of these traits.

Rotational methods may be either orthogonal or oblique. Oblique rotational methods do not maintain orthogonality between factors and therefore the results are not as easily interpretable. Oblique rotational methods are more prone to misuse (ref 12).

There are three common orthogonal rotation methods. Varimax rotation tries to maximize or minimize the loadings of the variables on each of the factors (columns). Quartimax tries to simplify rows (achieve non-trivial loadings for variables on only one factor). Equimax tries to simplify both rows and columns, something which is quite complex and difficult to perform sucessfully.

Previous Applications

No previous applications of factor analysis to computer

performance evaluation were discovered.

Approach

This technique is a major method of solving the problem of multicollinearity among the predictor variables. For this reason the data used in the chapter on Ridge regression wee used again. The index of multicollinearity for that data was known. The hypothesis was that factor analysis would be more sucessfully applied to data exhibiting a greater degree of multicollinearity. The reason for this hypothesis is that multicollinearity is an indication that the independent variables used are related to each other. The more related the variables are to each other the easier it should be to form cohesive factors. Assuming that the hypothesis is true, the implication is that this technique should be applied to data when multicollinearity is suspected, and any good factors uncovered should be used in place of the original variables in regression analyses in order to reduce the dimensionality of the model. The two primary methods of factor analysis (principal factor without iterations, also called principal component analysis, and principal factor with iterations) were both used and the results will be presented. Also used was Varimax rotation of factors with Kaiser normalization. The information presented in the Results section will apply to the rotated factors unless noted otherwise.

Results

As was expected, factor analysis was most useful on the data set showing the largest amount of multicollinearity among variables. It was also least sucessful on the data with the least amount of multicollinearity. These statements are based upon the the cummulative percentage of variation in the original variables that is explained by the factors, and on the rotated factors loadings, as shown in Table 7.1. Although it is reasonable to assume that the differences shown in the number of original variables between the three cases shown would partially account for the difference in the cummulative percentage of variation explained, there is another reason. Verification of this was performed by taking the same seven variables as used in the middle case, which was performed with data from data set 1, and using them in a Factor analysis of data set 2+ (the longer version of data set 2), which was used for the third case. Three factors were retained, explaining a total of 66.3% of the variance. The first two factors alone explained 51.1%. A significant reason for the differences in the amount of variance explained over the three cases is that the first two analyses included several variables which were largely uncorrelated with any other variables. The difference between the first and second cases is that two uncorrelated variables were removed for the second case and a single variable which was correlated to other variables, was added. The third case

Table 7.1

Three cases of Factor Analysis

Case Number	1	2	3
Data Set Used	1	1	2+
Multicollinearity	1.2	8.3	69.5
Number of Variables	8	7	4
Retained Factors	2	2	2
% Variation Expl.	43.0	57.0	89.0

Varimax rotated factor matrix after rotation with Kaiser normalization.

	Principal Components Factor Number		Principal Factors (w/iter.) Factor Number	
	1	2	1	2
Case 1				
tarriv	.1350	-.6048	-.0063	-.0884
cpu	.5659	.2576	.4351	.2524
mem	.6370	-.3271	.4692	-.0568
tapes	.7529	.0738	.6993	.1530
cards	.2642	.6574	.1459	.6424
lines	.5887	.0785	.4425	.1395
diskio	.3402	.4274	.2506	.2698
tapeio	.6297	.1912	.5152	.2125
Case 2				
iotime	.9553	.2357	.8746	.3539
cpu	.1737	.6184	.3786	.0943
mem	.0189	.6904	.3141	.0769
tapes	.3825	.6244	.5622	.0951
lines	.1280	.6591	.3521	.1289
diskio	.6140	.0529	.1853	.9656
tapeio	.8411	.2412	.8761	.0302
Case 3				
iotime	.9469	.3111	.9322	.3606
tapes	.0074	.8883	.0586	.5742
diskio	.9876	.0349	.9922	.0318
tapeio	.2140	.8694	.1521	.9754

made use of a different set of underlying data which was already more cohesive, and used only four variables, which were highly correlated with each other.

The identified factors exhibited higher loadings on those variables with which they were associated and were easier to name as the degree of multicollinearity increased. The factors also became more intuitively appealing. In reference to the first case, factor 1 was highly loaded on the number of tape drives, the amount of memory used, and the number of tape I/Os. Factor 2 had high loadings on arrival time and cards read. Neither of these factors is particularly easy to name or to grasp. In refrence to case two, the first factor was highly loaded on I/O time, tape I/Os, and disk I/Os, and the second factor was highly loaded on memory, lines printed, tape drives, and cpu time. Factor 1 was easily named "Input-Output," leaving "other resources" for the name of factor 2. In the third case (using data set 2+), factor 1 was highly loaded on I/O time and disk I/O. Factor 2 was highly loaded on the number of tape drives and of tape I/Os. The names for these two factors are obvious and the factors are appealing.

In Chapter 3 ridge regression was performed on data set 2+. The use of that technique resulted in great changes to the calculated regression coefficients, as had been expected for that highly multicollinear data. As shown in this chapter, the high degree of multicollinearity also allowed

strong factors to be found because the predictor variables were related to each other. This indicates that while these two techniques handle the problem of multicollinearity differently, they both were able to cope with it and provide evidence of its existence.

Recommendations

From all this, the conclusion is that factor analysis is of greater assistance in discovering common factors that allow reduction of the apparent dimensionality of the data in the case in which a fairly high degree of multicollinearity exits. If multicollinearity is suspected, the use of this technique will either deny the suspicion or show what variables can be combined into common factors. There may also be reasons for using factor analysis even when multicollinearity is not suspected in order to try to better understand the data being analysed. One caution is that the inclusion of a large number of variables into the analysis might obscure the existence of common factors by reducing their overall impact.

Chapter 8 Discriminant Analysis

Theoretical Considerations

As the name implies, Discriminant Analysis is used to discriminate between data cases, classifying them into two or more groups. It should be used when answers to the following questions are desired: (1) Do the available variables separate the populations?;(2) Can new observations be classified correctly?; and (3) Are the populations really different in the respect that their centroids are sufficiently distant from each other? (ref 12). This technique is different than the others discussed in this thesis so far because it requires data that is primarily nominally or ordinally scaled. The other techniques have assumed interval scaling. Discriminant analysis, unsurprisingly, gives the best results when it is used on data which consists of a set of disjoint groups which are characterized by low within groups sums of squared deviations (from the group centroid) and high between groups sums of squared deviations (from the overall centroid).

Typically, research using discriminant analysis proceeds in three steps. In step one the technique is used to find the discriminant function (of the available variables) that

used to test the null hypothesis tha
coincident versus the alternative h
separated. When there are more
Lambda can be used to test to s
centroids are separated.

If the second stage provides an a
third stage involves testing to se
variables can be in classifying the
correct groups. A classification fun
each group, and each observation to
evaluated using each function. The obser
into the group with the highest val
significance of the highest valued fu
indicates that the observation is close
centroid. Distances are measured in te
distance. This is similar to the normal E
except that the distance along each a
dividing by the standard deviation of
variable, so it is effectively a stand
measurement. Because classification of obser
function based upon their observed values is
the results, cross validation is often reco
consists of randomly choosing half the da
discriminant function and saving the other hal
classification function. A classification matr
showing the number of true observations in each

they were classified into each of the groups. Table 6.3 shows an example of a classification matrix. Notice that percentages of population sizes are shown beneath the actual numbers. Also shown is the overall percentage of correct classifications.

Previous Applications

No previous applications of discriminant analysis to computer performance evaluation were found.

Approach

Discriminant analysis has limited applications to computer performance evaluation because it works best when data is from several disjoint groups (nominally scaled) rather than from a single continuous population (intervally scaled). A proper application of discriminant analysis would be to discriminate between the four simulated organizations which submitted the computer jobs included in data set 1. However, this would not generally be helpful in computer performance evaluation because the imformation that could be obtained would normally already be available. Much more interesting would be the ability to discriminate between job turnaround time categories based upon the parameters available before the job was submitted. Reliable prediction of future turnaround times, by category at least, would be helpful both in the immediate sense for the submitters and in the CPE sense that improved turnaround times for a given set

of jobs would indicated improved system performance. The problem is that turnaround times, at least in the simulated data sets, show a continuous distribution pattern for which discriminant analysis yields poor results. The first example in the results section will show an example of an attempt to predict turnaround time categories.

A more realistic application of discriminant analysis involves discriminating between normal jobs and outlying jobs, that is jobs with extremely long turnaround times. Jobs with extremely short turnaround times cannot usually be considered outliers because there are often a great many of them. The second example in the results section shows an attempt to discriminate between normal and long jobs using data set 1.

Results

The SPSS select function was used to choose approximately half the 787 cases for use in calculating the discriminant function, leaving the remainder to test the classification function. The observations were also divided into three groups based upon observed turnaround times. Group 1 contained jobs with turnaround times of less than one-half hour, group 2 contained jobs with times of at least one-half hour but less than one and one-half hours, and group three contained the remaining jobs. The variables used in the discriminant and classification functions were the

independent predictor variables available (as presented in Chapter 1). A stepwise discriminant analysis was performed with the criterion being to maximize the Mahalanobis distance between the nearest pair of group centroids. Using Wilk's Lambda to test the null hypothesis that all the group centroids were coincident versus the alternative hypothesis that at least one group centroid was separate, it was found that the null hypothesis could be rejected at the .0001 level of significance (with .9999 confidence). Evaluation of the discriminant functions at the group centroids yeilded the following:

group	func 1	func 2
1	1.226	-.096
2	-.191	.143
3	-1.086	-.185

These results, primarily from function 1, show that there are clear differences between each of the three group centroids. equally spaced. The results of classification are shown in Table 8.1 for both the cases used in forming the discriminant and for the cases reserved. Notice that the former were classified slightly more accurately (59.6% to 57.1% correct). Also notice that group 1 and 3 observations were classified fairly well but that group 2 observations were spread between all three groups. This is due to the continuous nature of the data distribution. A note that should be made for serious readers is that Box's M statistic showed that the group covariance structures were not equal.

Table 8.1

Classification Matrices for example 1

Results for cases selected for use in the analysis

Actual Group	Number of cases	Predicted Group Membership 1	2	3
1	107	85 79.4	18 16.8	4 3.7
2	186	48 25.8	84 45.2	54 29.0
3	88	5 5.7	25 28.4	58 65.9

59.58 Per cent grouped correctly

Results for cases not selected for use in the analysis

Actual Group	Number of cases	Predicted Group Membership 1	2	3
1	121	95 78.5	23 19.0	3 2.5
2	192	53 27.6	79 41.1	60 31.1
3	93	4 4.3	31 33.3	58 62.4

57.14 Per cent grouped correctly

This is a violation of one of the assumptions of discriminant analysis and should be kept in mind when evaluating the results.

The second analysis run was an attempt to discriminate between outliers and normal jobs in terms of turnaround time. Plotting turnaround times for the 787 jobs in data set 1 showed that outliers had turnaround times of more than 3.2 hours. Up to that point the distribution of jobs was fairly continuous. Making use of this information, a discriminant analysis was performed. Of the 787 cases, 9 (about 1%) were identified as outliers. The variables representing the number of tape drives allocated and the amount of memory required proved the best at discriminating between the groups. For both variables the mean values were more than twice as large for the outlier group as for the normal group. The mean values of each of the variables for the two groups is shown in Table 8.2. The number of tape accesses and the amount of cpu time used also assisted in the discrimination. No other variables proved significant in this regard.

A test of the significance of the difference between the two group centroids proved significant at the .0001 level, therefore the null hypothesis that the group centroids were coincident was rejected. The discriminant function coefficients and values of the discriminant functions evaluated at group centroids are shown in Table 8.3. Because there were only 9 cases in the outlier group, no attempt to

Table 8.2

787 Computer jobs grouped by turnaround time

Group		Normal	Outliers
Number of cases		778	9
Mean of vars:	tapes	1.39	2.78
	mem	68.2	163.6
	tapeio	232.7	287.7
	cpu	154.4	167.7
	diskio	893.4	1073.4
	tarriv	12.1	11.7
	cards	238.8	256.9
	lines	1042.5	1483.0

The predictor variables used are:

cpu	the amount of cpu time used by a job (seconds)
mem	the amount of central memory used (k-bytes)
cards	the number of cards read in
lines	the number of lines of output printed
diskio	the number of disk accesses made
tapeio	the number of tape accesses made
tapes	the number of tapes to be mounted concurrently
tarriv	the time when the first card of a particular job is read in (expressed in decimal hours)

Table 8.3

Discriminant Function Coefficients

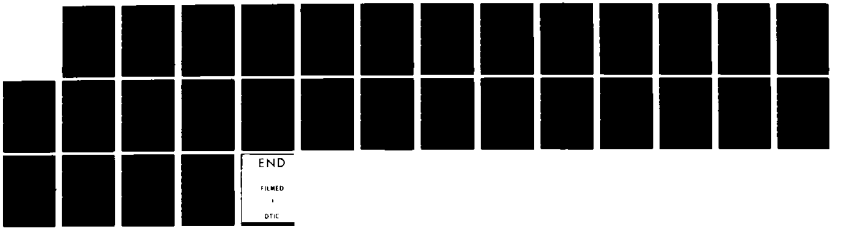
Unstandardized Canonical Discriminant Function Coefficients

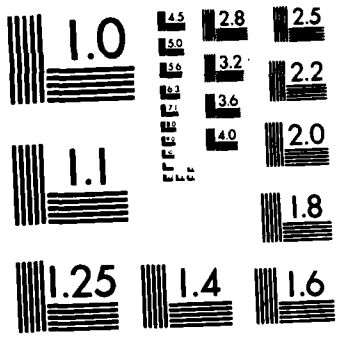
tapes	.4544952
tapio	-.0007441
cpu	-.0011345
mem	.0166141
constant	-1.442342

Canonical Discriminat Functions Evaluated at Group Centroids

Group	Function
Normal	-.02472
Outliers	2.13691

AD-A124 899 A STUDY OF MULTIVARIATE STATISTICAL ANALYSIS TECHNIQUES 2/2
FOR COMPUTER PERF..(U) AIR FORCE INST OF TECH
WRIGHT-PATTERSON AFB OH SCHOOL OF ENGI.. G MAGAVERO
UNCLASSIFIED DEC 82 AFIT/GCS/EE/82D-23 F/G 12/1 NL





cross validate the classification function was attempted. The classification function was able to correctly place 85.51 per cent of the observations. This seems quite good but 113 observations from the normal group were misclassified into the outlier group. This greatly exceeds the actual number in that group, which numbered only nine. Eight of the nine were correctly classified. These results are shown in Table 8.4. The results are both good and bad. The bad part is that 113 jobs were incorrectly predicted to be extremely long, but were not. The good part is that those jobs that were extremely long were also predicted to be so. It is also suspected that of the 113 jobs incorrectly classified, most would have been near the upper end of the distribution of normal jobs in terms of turnaround time and therefore their classification as outliers could be used by the submitter as a warning that they may take an extemely long time to run.

Another thing to note is that, again, the distribution of turnaround times is fairly continuous and therefore this is putting discriminant analysis to a test on which we should not expect it to perform well.

Recommendations

It does not seem that discriminant analysis should be generally used in the analysis of computer performance data, but that it should be used only in well selected cases in which the data may reasonably be placed into discrete

Table 8.4

Classification Function Coefficients and Results

Classification Function Coefficients (Fisher's Linear Discriminant Functions)

Group =	Normal	Outliers
tapes	.7199	1.7024
tapeio	-.5347E-04	-.1662E-02
cpu	.1453E-02	-.9991E-03
mem	.2146E-01	.5737E-01
constant	-2.0321	-7.4328

Classification Results

Actual Group	Number of cases	Predicted Group Membership Normal	Outliers
Normal	778	665 85.5	113 14.5
Outliers	9	1 11.1	8 88.9

85.51 Per cent of all cases classified correctly

groups. If it is suspected that the population consists of discrete sub-groups the use of this technique could confirm or deny the suspicion. A high percentage of correct classifications would act as confirmation.

Chapter 9. Conclusions and Recommendations

This thesis has investigated the application of several multivariate statistical techniques to the analysis of computer performance data. These techniques were Ridge Regression, Automatic Interaction Detection, Cluster Analysis, Canonical Correlation Analysis, Factor Analysis, and Discriminant Analysis. Ordinary least-squares multiple linear regression analysis was used as a baseline against which to compare the other procedures as applicable. Data used in the analysis was obtained from a computer simulation developed at the Air Force Institute of Technology as a prior masters thesis project. Several data sets, representing a variety of complexities, were used.

The basic concerns of computer performance evaluation are system throughput, job turnaround time, and, for interactive systems, response time. The applications made for each technique were designed to take into account these concerns as well as the intended use of the technique and their theoretical requirements and capabilities.

Three main data sets were used in this thesis. Data set 1 was from a realistic simulated computer environment, with a great deal of complexity and competition for resources. Data set 2 (and its longer version, data set 2+) was produced in

an extremely non-competitive environment. The data is simple, easy to analyze, and exhibited a fairly high degree of multicollinearity. Data set 3 was a small data set designed to be an intermediate step between the the other two sets. Its characteristics are also intermediate between the other two.

Ridge Regression

Ridge regression is primarily used to counteract the effect of multicollinearity between variables upon the calculated regression coefficients. For this reason it was applied to two data set pairs, each of which consisted of two different combinations of predictor variables. In one case, in each set, the degree of multicollinearity between the variables was low and in the other case certain variables were added or deleted so that the multicollinearity index was moderate or high.

The first data set pair was a contrived example of two orthogonal predictor variables plus a third predictor variable that was almost perfectly correlated with the second. This data was not one of the main sets mentioned above, but a data set created specifically to demonstrate the effects of multicollinearity and ridge regression. The addition of the third variable caused a drastic change in the regression coefficient calculated for the second variable. The use of the ridge technique was largely able to restore

the coefficient to its original value. A very similar process was performed with the second data set pair, which consisted of simulated computer performance data. The results achieved were also similar.

The ridge technique adds a small amount of bias into the calculation of regression coefficients. It is hoped that the added bias will lead to calculated coefficients which are more stable and which allow a better interpretative view of the contributions of each predictor variable. Several heuristics for determining the amount of bias to add into the calculations were considered. Two of these performed approximately equally well. These are first, all signs of coefficients must be correct (changed to their final value), and second, all Variance Inflation Factors (VIFs) must be no more than 1.0 (as would be the case for orthogonal variables). VIFs are a measure of the relatedness of the predictor variables. The recommendation of this thesis is to choose the one of these that requires the addition of less bias.

Automatic Interaction Detection

Automatic interaction detection (AID) was proposed by its authors, J.A. Sonquist and J.N. Morgan, as a method for automating the search for structure within data. Specifically, one of its abilities is supposed to be the ability to detect interaction between variables. The technique works by

splitting a large (at least 1000 observations with outliers on the criterion variable removed is Sonquist's recommendation) data set in such a way that each split produces the greatest possible reduction in within groups variation. The possible splits are predefined by the user at intervals on the predictor variables. At each iteration, the split is chosen by performing a one-way analysis of variance for each possible split. This iterative process continues until one of several stopping criteria is reached. The final result is a tree structure along with descriptive statistics on each of the terminal subgroups. Signs of interaction among variables are asymmetric tree structure and increases in the amount of reduction in within groups variation possible through splitting on a particular variable after the effects of another variable is removed by the splitting process.

For this thesis, AID was applied to a simple data set and a more complex data set. The simple data set produced a symmetric tree and showed no signs of interaction. This was as expected. The complex data set produced an assymetric tree structure and the amount of within groups variation reduction possible by splitting on one of the variables, the number of tape drives, increased substantially (31.9% and 46.9%) after splits were performed on other variables. These are both signs of interaction. The interacting variables were confirmed by adding product terms to a stepwise regression analysis. Two product terms proved to be highly

significant, and their inclusion improved the R-squared statistic almost 20% (.424 to .505).

While these interaction terms could have been found by using a trial and error process, this method was more structured and was probably less time consuming. A real limitation on the use of AID is that data must be in integer form.

Cluster Analysis

Cluster analysis is similar to AID, but the two techniques differ importantly. AID groups observations based upon similarity in the value of the criterion variable. Cluster analysis groups data based upon the values of all variables included.

A great variety of clustering algorithms exist. The interested reader is directed to Anderberg's book (ref 3). This is an extensive and pleasing discussion of clustering. Cluster analysis is normally performed as the first step in an analysis, rather than a complete investigation in itself. In this regard there are many applications of the technique to computer performance evaluation. Cluster analysis has been used in previous studies to form clusters, each of which could then be analysed independently with other techniques. In this way the confounding effects of the other clusters and the space between the clusters could be removed. The work done for this thesis is similar to work done by T.C.

Hartrum and J.W. Thompson (ref 9).

For this thesis, data was clustered both before and after the cpu round-robin time quantum of the simulated computer was increased. Because the data was simulated, the same jobs could be submitted both before and after the change, therefore exactly the same clusters were formed both before and after. The effect of the time quantum change was determined by the change in the mean turnaround time for observations in each of the clusters. While some turnaround times were increased after the change, others were decreased. A logical next step in a real data analysis would be to tie together the factors that caused some times to increase and others to decrease. However, this is not required to demonstrate the use of cluster analysis.

Cluster analysis is recommeded for the analysis of data that is widely dispersed or about which little is known concerning the data's structure. It can show the structure of the data or allow the formation of improved models through the removal of confounding effects.

Canonical Correlation Analysis

Canonical correlation analysis is used to relate factors, represented by canonical variates, each of which is described by a set of logically related variables. Dependency of one canonical variate upon the other is not necessary, but is allowable. This technique can reduce the

apparent dimensionality of the data to a comprehensible level while allowing retention of a dependency relationship if desired. This technique has been used extensively to relate attitutes to measureable responses on surveys.

Because this technique is used to both reduce dimensionality and relate factors, it makes sense to apply it when there are a large number of variables to be related. For the example performed for this thesis, computer job size, measured by turnaround time and I/O time, was related to a factor representing resource utilization. This was performed on simulated data produced in both a simple environment and in a more complex environment. In the complex case, the two measures of job size related to different utilization variables. Therefore it would have been more appropriate to have performed two separate linear regression analyses. In the case of the simple data, the two measures of job size were more closely related to each other and were related to the same resource utilization variables.

The conclusion reached was that this technique should be applied to computer performance analyis only when there are a large number of performance variables and when the variables in each set are closely related.

Factor Analysis

Factor analysis is also used to reduce the apparent dimensionality of data. It, however, attempts to do this by

finding factors or latent variables which adequately convey the information available from the manifestation (measureable) variables. This is helpful when there are too many variables to allow comprehension of the role of each and when the logical relationships between variables is unknown. This technique is particularly helpful if the variables are multicollinear.

Each of the several available variations of this technique determines a number of factors equal in number to the number of manifestation variables. The factors are defined to be orthogonal to each other and are ordered by their eigenvalue (a measure of the relative amount of variation of the data explainable by that factor). The number of factors to be retained is reduced based upon some criterion such as "each factor must be able to explain at least as much of the variation of the data as an original manifestation variable." Normally the retained factors are rotated in an attempt to make them correlate highly (have high loadings on) some of the manifestation variables, and very low loadings on the others. It is also desired that each variable have a non-zero loading on only one factor. In this way the relationship between factors and the manifestation variables can be seen and, possibly, the factors can be named.

This study used two variations of factor analysis: principal component analysis (also known as principal factor

analysis without iterations) and principal factor analysis with iterations. Varimax rotation and Kaiser normalization were also used in both cases. Three data sets, exhibiting low, medium, and high degrees of multicollinearity were analysed with these techniques. Factors explaining the greatest percentage of variation in the data were obtained for the data with the highest degree of multicollinearity. Neither variation of factor analysis seemed to be clearly superior to the other in terms of being able to get clearly high and clearly low loadings on the variables, and in getting the variables to have non-zero loadings on only one factor.

The conclusion drawn from this analysis was that this technique is most appropriately used on data exhibiting a high degree of multicollinearity. In a sense, this technique can be used to confirm or deny suspicions of multicollinearity because multicollinear data will form factors which are able to explain a large amount of variation in the data. A caution in using this technique is that the inclusion of a large number of unrelated variables in the model may obscure the existence of common factors by reducing their impact.

Discriminant Analysis

Discriminant analysis is used to determine if the data is composed of discrete groups (with regard to the variables used) and if new observations can be correctly placed into

these groups. This technique is the only one used for this thesis that requires the data to be nominally or ordinally scaled. The other techniques all assumed interval scaling.

The analysis normally procedes in three steps. The first step is determination of the discriminant function that produces the greatest separation of the user proposed groupings within the data. The second step is to test the significance of the difference between the group centroids. If a significant difference is found between at least two groups, the third step is to classify data observations with known group affiliation and to determine how successful the classification is.

Because it would be desirable in computer performance analysis to predict the turnaround time of a computer job, within class limits at least, discriminant analysis was used in an attempt to do this. It must be noted that this is not an appropriate test of the technique because job turnaround times fall into a continuous distribution rather than into the discrete groups required for this technique. Indeed, the results showed this to be true. It was not possible to clearly discriminate between three job turnaround time intervals. The jobs in the middle interval substantially overlapped the extreme intervals even though the group centroids were significantly different (at the .0001 level).

A slightly more appropriate example of the use of discriminant analysis was an attempt to predict membership in

only two groups: jobs with normal turnaround times and those with extremely long turnaround times (outliers). Although eight of the nine outliers were correctly classified, 113 of the normal jobs were misclassified as outliers. This resulted in an overall sucessful prediction rate of only about 85%. This was viewed as a mixed sucess because although so many normal jobs were misclassified, they were presumably near the upper end of the distribution of normal jobs in terms of turnaround time. Therefore, even this misclassification would probably be useful in computer performance analysis, provided, of course, that in an actual study of data this presumption would be verified.

This technique should probably not be routinely applied to the analysis of computer performance. Instead, it should be used when it is suspected that discrete groups exist within the data. The use of this technique can confirm or deny this suspicion.

Comparison of Techniques

As shown in Table 9.1, all the techniques could be applied to computer performance evaluation, but the application of discriminant analysis was strained because the data was intervally scaled. The techniques, in general, have uses which differ from ordinary least squares linear regression analysis, so a direct comparison of results is not realistic. As a general trend, however, the simpler and more

Table 9.1 (Part 1)

Comparison of Results

	Ridge Regression	AID	Cluster Analysis
application	removed effect of multicol-linearity from calculated regression coefficients.	examined data for structure and quadratic terms.	determined effects on clusters when system param-eter changed.
result with data set 1	coefficients of related variables changed considerably.	discovered a quadratic term.	showed that different clusters had different reactions to syst. change.
data set 2	coefficients changed considerably.	showed sym-metry and no interaction.	not used
data set 3	not used	not used	not used
technique applicable?	yes	yes	yes
when?	data shows multicollin-earity and explanatory coefficients required.	to find structure of data and in-teraction terms.	to find structure of data as first part of study or to study indiv. groups.

Table 9.1 (Part 2)

Comparison of Results

	Canonical Correlation	Factor Analysis	Discriminant Analysis
application	related job length to job resource use.	found factors to reduce data dimensionality.	tried to classify turnaround time categories.
result with data set 1	measures of job length not dependent on the same variables.	showed weak factors.	not able to classify by time categories, and only somewhat for normal vs. outliers.
data set 2	measures of job length were closely related and dependent on same variables.	showed good factors.	not used.
data set 3	measures of job length not dependent on same variables.	not used.	not used.
technique applicable?	sometimes	sometimes	not normally.
when?	when relating sets of vars. and to reduce dimensionality.	to reduce dimensionality.	only if data is nominal or ordinal scaled.

cohesive the data, the better regression analysis did, and the better the other techniques did. AID provided an exception to this generalization because its purpose is to determine the complexity of the data, which it seemed to do well both for simple and complicated data.

Ridge regression, canonical correlation analysis, and factor analysis are best applied when the variables are somehow related to each other. Ridge regression helps to remove the effects of these relationships from the calculated regression coefficients. The other two techniques exploit the relationships to find a dimension-reduced representation of the data. Canonical correlation analysis relates two groups of variables and can maintain a dependency relationship. Factor analysis works with only one set of data, but can rotate factors to get better results.

Results from the use of ridge regression showed that when a substantial amount of multicollinearity was present in the data, the regression coefficients changed greatly as the amount of bias entered into the calculation was increased. The predictivity of the model, as measured by the R-squared statistic, also dropped off significantly as the coefficients changed. Therefore, if predictivity is the main concern, ordinary least squares regression analysis is superior for data with extreme multicollinearity, however ridge regression is superior if an explanatory model is desired. For data with little multicollinearity there was no great effect

caused by the introduction of bias into the calculations. The ability of factor analysis to uncover strong factors exhibited the same trend that ridge regression did in producing significantly changed coefficients. The most striking results were obtained with the most highly multicollinear data. The results achieved with canonical correlation analysis followed the same trend, producing cohesive, related factors for the highly multicollinear data.

AID, cluster analysis, and discriminant analysis all work with population subgroups. AID and cluster analysis produce the subgroups, while discriminant analysis attempts to determine if the user suggested subgroups really exist as separate entities in the data. AID clusters observations based upon similarity in the value of the criterion variable, while cluster analysis takes account of similarity of all variables when grouping observations. Additionally, AID may discover interaction between variables.

Because AID and cluster analysis are intended to help to determine the characteristics of data, it would be expected that the successfulness of their use would not be dependent upon such characteristics, and indeed it did not appear to be. Of course, the cohesiveness of clusters formed would be dependent upon the relatedness of the data, but finding that the data is not closely related should not be considered a failure of the technique. This is also true of discriminant analysis. In this thesis, it was put to tests for which it

was not designed to produce good results, and it didn't. This is not a failure of the technique but of the attempted use.

Learning Experiences

The most significant learning experience involving this thesis was in finding first hand how difficult it is to produce useable data files from real computer accounting information. The reduction in volume of information in going from raw individual job accounting report messages to useable data on the Cyber computer was from 50,000 card images to 39 data records, most missing some data. A misunderstanding of the purpose of ridge regression was caught and corrected by members of the thesis committee. Much was also learned about cluster analysis from Anderberg's book (ref 3).

Recommendations for Further Study

Because researchers tend to be skeptical of analyses done using simulated computer data (and often with good cause), the most significant recommendation for further study is to apply these techniques to several different sets of real computer accounting data. It would be preferable if this study could be done as a real CPE project and actual findings could be reported as well as the usefulness of each of the techniques in solving particular problems. The study should also show how the techniques can be used in an

integrated approach to computer performance analysis.

Further study in the area of computer simulation would also be warranted. Using these techniques in combination with computer simulations such as CPESIM, researchers could change the computer system environment and correlate it with the usefulness of the various techniques.

A final recommendation is to consider studying any other known or potential application of these techniques to computer performance evaluation.

Bibliography

1. Afifi, A.A. and S.P. Azen. Statistical Analysis - A Computer Oriented Approach. New York: Academic Press, 1979.

2. Agrawala A.K. and J.M. Mohr. "Some Results of the Clustering Approach to Workload Modelling," Proceedings of the Thirteenth Meeting of the Computer Performance Evaluation Users Group, 23-38, 1977.

3. Anderberg, Michael R. Cluster Analysis for Applications. New York: Academic Press, 1973.

4. Bennett, Spencer and David Bowers. An Introduction to Multivariate Techniques for Social and Behavioral Sciences. New York: Halsted Press, John Wiley and Sons, 1976.

5. Chatterjee, Samprit and Bertram Price. Regression Analysis by Example. New York: John Wiley and Sons, 1977.

6. Department of Operational Sciences. "Preliminary Description: RIDGE, A Program to Support Ridge Regression Analysis," notes. School of Engineering, Air Force Institute of Technology, Wright-Patterson A.F.B., Ohio, undated.

7. Gomaa, H. "A Modelling Approach to the Evaluation of Computer System Performance," 171-179. Modelling and Performance Evalution of Computer Systems. North Holland Publishing Company, 1976.

8. Gooch, L.L. Policy Capturing with Local Models: The Application of the Automatic Interaction Detection Technique in Modeling Judgment, unpublished Ph. D. dissertation. University of Texas, Austin, Texas, 1972.

9. Hartrum, Thomas C. and Jimmy W. Thompson. "The Application of Clustering Techniques to Computer Performance Modelling," Proceedings of the 15th Meeting of the Computer Performance User's Group, October 1979.

10. Hartrum, Thomas C. Lecture Materials distributed in EE752, Computer Performance Evaluation. School of Engineering, Air Force Institute of Technology, Wright-Patterson A.F.B, Ohio, 1982.

11. Marathe, Madhav. "A Statistical Comparison of the System Performance of Several Configurations," 227-238. Proceedings of the Fourteenth Meeting of the Computer Performance Evaluation Users Group, October 1978.

12. McNichols, Charles W. An Introduction to: Applied Multivariate Data Analysis, unpublished text. School of Engineering, Air Force Institute of Technology, Wright-Patterson A.F.B., Ohio, 1980.

13. McNichols, Charles W. and James R. Makin. "A Monte Carlo Investigation of the Applicability of Ridge Regression to Developing Cost Estimating Relationships," Paper presented at the 1982 spring Joint Conference of The Operations Research Institute of Management Sciences and the Operations Research Society of America, Detroit, Michigan, April 1982.

14. Nie, Norman H. et al. SPSS Statistical Package for the Social Sciences. Users manual. New York, McGraw-Hill Book Company, 1975.

15. Stover Margaret A. Application of Statistical Analysis Techniques to Computer Performance Evaluation, Masters thesis. Air Force Institute of Technology, Wright-Patterson A.F.B., Ohio, 1981.

Appendix A

Computer Performance Evaluation Simulator (CPESIM)

CPESIM is a set of computer programs created by Paul Lewis as a Masters thesis effort at the Air Force Institute of Technology, and modified slightly by T.C. Hartrum for use in CPE classes there. It consists of two parts. The first part is a simulated computer system written in Simscript. The second part is a user alterable operating system specification. This consists of a workload specification and a resource specification. The workload specification is used to generate a set of user jobs called the event stream. The event stream may be generated using randomized generation procedures (with user selected distributions and parameters) or this procedure may be bypassed by using actual computer performance data. Workload statistics include arrival time and day, cpu time, central memory, priority, number of tape drives, numbers of cards read, lines printed, tape I/Os, and disk I/Os. The resource specifications include multiprogramming level, cpu time quantum for round robin, number and size of fixed memory partitions, type (relative speed) of cpu, number and speed of disk drives, drums, tape drives, and I/O

controllers. Also included are the number and speed of card readers and line printers. Hardware and software monitors may also be requested by including them in the resource specification.

Each job, including system jobs, requires a memory partition and competes for the use of the cpu and I/O channels and devices. Tape drives and memory are allocated when a job is loaded into core and are not released until the job is complete. I/O devices are allowed direct memory access to central memory.

All input jobs are assumed to be on cards. Turnaround time includes the time from when the cards are placed into the card reader until the last line of output is printed (or until the last I/O or cpu execution is complete if no lines are printed). Jobs are read in by the input spooler, which requires a 4K partition and a fixed amount of cpu time. The job scheduler, which is core resident, loads jobs into memory and allocates required tape drives. The process scheduler selects jobs from the execute queue to run on the cpu. Execution time is limited to the shorter of the time quantum and the time between I/O operations. Jobs requiring additional cpu time are returned to the rear of the execute queue when they are again ready (after they complete any required I/O).

Data transfers are made in fixed 1K byte blocks. If either the device or the channel is busy, the job is placed

on the device queue. All queues are served in a FIFO (FCFS) manner. The I/O scheduler controls all I/O. The output spooler prints the output lists of all completed jobs. It is nearly identical to the input spooler.

In a student situation, the instructor defines all workload specifications which are then submitted to a workload generator. This outputs a set of individual jobs which form the event stream. Different event streams can be formed by altering the workload specifications or the random number seed on the workload generator. The instructor also defines a baseline computer configuration which has some bottlenecks in it. Students receive the outputs from several simulations. These are analysed using the techniques taught in the course and prerequisite statistical analysis courses to determine where the bottlenecks are located. The students may then alter the software and/or hardware configuration and rerun the simulation. The new data is analysed to determine if the changes made were beneficial.

Appendix B - Ridge Program

```
      PROGRAM RIDGE(INPUT,OUTPUT,TAPE8=INPUT,TAPE9=OUTPUT)
C     PROGRAM TO PERFORM RIDGE REGRESSION ANALYSIS
C     C.NICHOLS -- MAY 1980 -- AIR FORCE INSTITUTE OF TECHNOLOGY
C     PROCEDURE AS DOCUMENTED IN CHATTERJEE AND PRICE
C     "REGRESSION ANALYSIS BY EXAMPLE" WILEY,1977
C     MULTIPLE REGRESSION PROCEDURE IS BERNSMSOM'S ALGORITHM
C     IN "MATHEMATICAL METHODS FOR DIGITAL COMPUTERS" ED. BY
C     RALSTON AND WILF WILEY,1960
C     DATA BASE FORMAT:
C     FIRST CARD -- COLS 1-2 NUMBER OF VARIABLES (INCLUDING DEPENDENT)
C                   LIMIT IS 16. COLS 3-4 INDEX OF DEPENDENT VAR.
C                   COLS 5-6, ANY NON-ZERO VALUE GENERATES
C                   LOG-LINEAR MODEL. COLS 7-11 K-INCREMENT VALUE,
C                   MUST BE GREATER THAN ZERO AND LE .02.
C     SECOND CARD -- FORTRAN FORMAT STATEMENT FOR INPUT DATA. MUST BE
C                    F-TYPE SPECIFICATIONS AND ACCOUNT FOR NUMBER OF
C                    VARIABLES STATED ON FIRST CARD
C     REMAINING CARDS -- OBSERVATIONS IN FORMAT SPECIFIED BY SECOND CARD
      DIMENSION A(16,16),R(16,16),M(16),S(16),X(16),B(50,16),PLT(50,52),
     1       FMT,8),ALNUM(16),FKMX(51),VIF(50,16)
C     CORRELATION MATRIX CONSTRUCTED IN A(,)
C     CORRELATION MATRIX COPIED TO R(,) FOR EACH ITERATION
C     M() IS MEAN VECTOR, S() IS STD.DEV. VECTOR
C     B(,) IS MATRIX OF STANDARDIZED COEFFICIENTS
C     PLT(,) IS PLOT BUFFER FOR RIDGE TRACE; PLT(I,52) IS R-SQUARE
C     FKMX() CONTAINS VALUES OF K FOR EACH ITERATION
C     VIF(,) CONTAINS VARIANCE INFLATION FACTORS FOR EACH ITERATION
      REAL M,INCK
      DATA ALNUM/"1","2","3","4","5","6","7","8","9",
     1  "A","B","C","D","E","F","G"/
      N=0
C     LOAD DATA BASE. FIRST READ NO. VARS:NV, INDEX OF DEPENDENT: IXD
C     FLAG FOR LOG-LINEAR: LOGF, INCREMENT FOR K-VALUE: INCK
      READ(8,10) NV,IXD,LOGF,INCK
10    FORMAT(3I2,F5.3)
      IF(INCK.LE.0.0) INCK=.005
      IF(INCK.GT..02) INCK=.005
      NVM1=NV-1
C     INITIALIZE
      DO 100 I=1,NV
       M(I)=0.0
       S(I)=0.0
       DO 100 J=1,NV
        A(I,J)=0.0
100   CONTINUE
      DO 150 I=1,50
       DO 150 J=1,16
        B(I,J)=0.0
150   CONTINUE
C     READ FORMAT STATEMENT DESCRIBING DATA BASE
      READ(8,15) FMT
15    FORMAT(8A10)
C     READ OBSERVATIONS ACCORDING TO USER INPUT FORMAT STATEMENT
200   READ(8,FMT) (X(J),J=1,NV)
      IF(EOF(8).NE.0) GO TO 400
      IF(LOGF.EQ.0) GO TO 250
      DO 225 J=1,NV
       X(J)=ALOG(X(J))
225   CONTINUE
250   N=N+1
C     CONSTRUCT MEAN VECTOR AND COVARIANCE MATRIX
      DO 300 J=1,NV
       M(J)=M(J)+X(J)
       DO 300 J1=J,NV
```

```
          A(J,J1)=A(J,J1)+X(J)*X(J1)
3ØØ     CONTINUE
C       NEXT CASE
        GO TO 2Ø
C       END OF INPUT DATA, CALCULATE MEANS,SIGMAS,CORRELATION MATRIX
4ØØ     DO 5ØØ J=1,NV
         S(J)=SQRT((A(J,J)-M(J)*M(J)/N)/(N-1.Ø))
         M(J)=M(J)/N
5ØØ     CONTINUE
        DO 6ØØ J=1,NV
         DO 6ØØ J1=J,NV
          A(J,J1)=(A(J,J1)-N*M(J)*M(J1))/((N-1.Ø)*S(J)*S(J1))
6ØØ     CONTINUE
        DO 7ØØ J=1,NVM1
         JP1=J+1
         DO 7ØØ J1=JP1,NV
          A(J1,J)=A(J,J1)
7ØØ     CONTINUE
C       PRINT MEANS,STD.DEVIATIONS,CORRELATION MATRIX
        WRITE(9,3Ø) INCK,N
3Ø      FORMAT(1H1,"RIDGE REGRESSION PROGRAM -- AIR FORCE INSTITUTE OF"
       1 " TECHNOLOGY"/1HO,"K-VALUE INCREMENT IS ",F6.4///
       2 1HO,I8," CASES READ FROM INPUT FILE"//
       3 1HO,"VARIABLE NUMBER       MEAN       STD.DEV."/)
        DO 8ØØ J=1,NV
        WRITE(9,35) J,M(J),S(J)
35      FORMAT(1H ,7X,I2,6X,F12.5,F12.4)
8ØØ     CONTINUE
        IF(LOGF.EQ.Ø) GO TO 85Ø
        WRITE(9,37)
37      FORMAT(1HØ/1HØ,"LOG-LINEAR OPTION, ALL VARIABLES TRANSFORMED"/)
85Ø     WRITE(9,4Ø) (NN,NN=1,NV)
4Ø      FORMAT(1HØ/1HØ,"CORRELATION MATRIX"/1HØ,"VARIABLE",
       1   16I7)
        DO 9ØØ J=1,NV
        WRITE(9,45) J,(A(J,J1),J1=1,NV)
45      FORMAT(1HØ,I6,4X,16F7.3)
9ØØ     CONTINUE
C       COPY CORRELATION MATRIX FROM A TO R FOR EACH ITERATION
C       FK IS VALUE OF K FOR RIDGE ESTIMATES
        FK=Ø.Ø
        FKMX(1)=Ø.Ø
        WRITE(9,5Ø) (NN,NN=1,NV)
        WRITE(9,52)
5Ø      FORMAT(1H1,"NORMALIZED (STANDARDIZED) REGRESSION COEFFICIENTS"/
       1 1HØ,"  VARIABLE:",16I7)
52      FORMAT(1H ,"K-VALUE")
        DO 165Ø IXK=1,5Ø
         DO 1ØØØ J=1,NV
          DO 1ØØØ J1=1,NV
           R(J,J1)=A(J,J1)
1ØØØ     CONTINUE
C       ALTER DIAGONAL OF R MATRIX REPRESENTING X'X
        DO 11ØØ J=1,NV
         IF(J.EQ.IXD) GO TO 11ØØ
         R(J,J)=R(J,J)+FK
11ØØ    CONTINUE
C       MATRIX INVERSION -- SOLVES FOR REGRESSION COEFFICIENTS
        DO 15ØØ I=1,NV
         IF(I.EQ.IXD) GO TO 15ØØ
         DO 13ØØ J=1,NV
          IF(J.EQ.I) GO TO 13ØØ
          V=R(J,I)/R(I,I)
          DO 12ØØ K=1,NV
```

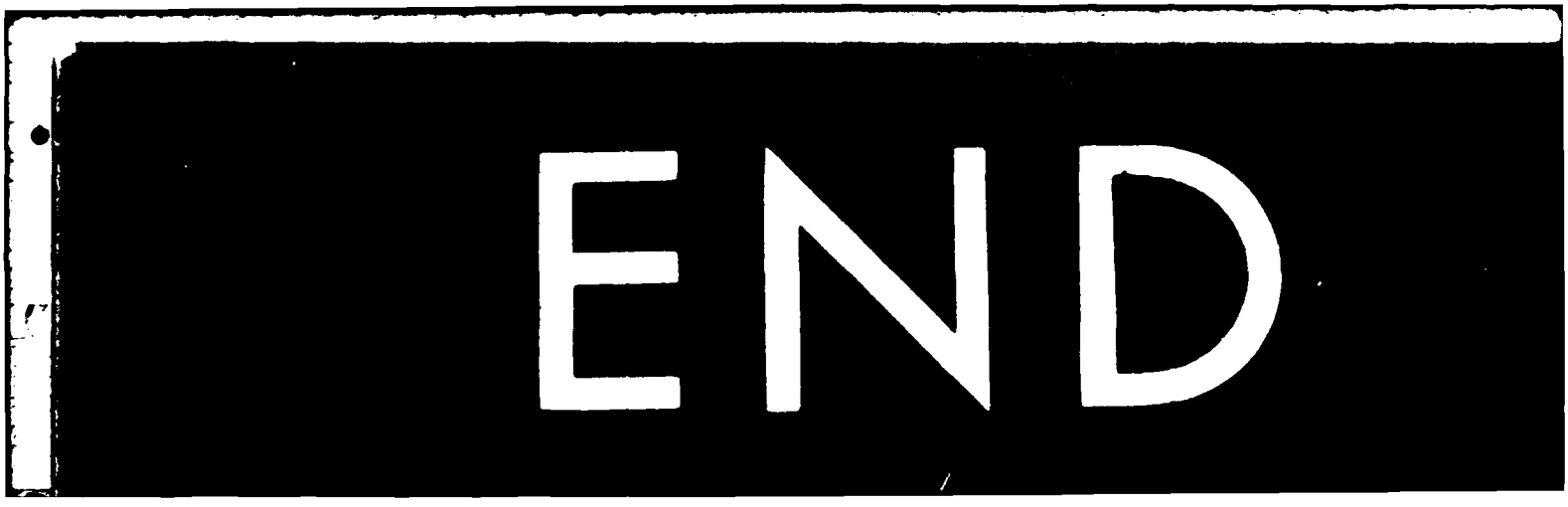
```
          IF(K.EQ.I) GO TO 12ØØ
          R(J,K)=R(J,K)-V*R(I,K)
12ØØ     CONTINUE
         R(J,I)=-V
13ØØ     CONTINUE
         DO 14ØØ K=1,NV
          IF(K.EQ.I) GO TO 14ØØ
         R(I,K)=R(I,K)/R(I,I)
14ØØ     CONTINUE
         R(I,I)=1.Ø/R(I,I)
15ØØ     CONTINUE
C        SAVE COEFFICIENTS FROM THIS ITERATION
C        CALCULATE VIF'S AND SAVE:
C        DIAGONAL ELS OF COEFFICIENT COVAR. MTX. DIVIDED BY SIGMA**2
         BSQ=Ø.Ø
         DO 16ØØ J=1,NV
          VIF(IXK,J)=Ø.Ø
          IF(J.EQ.IXD) GO TO 16ØØ
          B(IXK,J)=R(J,IXD)
          BSQ=BSQ+B(IXK,J)*B(IXK,J)
          DO 1575 L=1,NV
           TVIF=Ø.Ø
           IF(L.EQ.IXD) GO TO 1575
           DO 155Ø K=1,NV
            IF(K.EQ.IXD) GO TO 155Ø
             TVIF=TVIF+A(L,K)*R(K,J)
155Ø      CONTINUE
          VIF(IXK,J)=VIF(IXK,J)+R(J,L)*TVIF
1575      CONTINUE
16ØØ     CONTINUE
C        SAVE R-SQUARE VALUE IN PLOT BUFFER: B'X'Y+K*B'B
         PLT(IXK,52)=1.Ø-R(IXD,IXD)+FK*BSQ
C        END OF LOOP OVER VALUES OF K
C        PRINT COEFFICIENTS FOR THIS ITERATION
         WRITE(9,55) FK,(B(IXK,J),J=1,NV)
55       FORMAT(1H ,F5.3,6X,16F7.3)
C        ALTER K VALUE
         FK=FK+INCK
         FKMX(IXK+1)=FK
165Ø     CONTINUE
C        CALCULATE UNNORMALIZED COEFFICIENTS
         WRITE(9,51) (NN,NN=1,NV)
51       FORMAT(1H1,"UNNORMALIZED COEFFICIENTS"/
        1 1HØ," VARIABLE:INTERCEPT  ",I8,9I11)
         WRITE(9,52)
         DO 18ØØ I=1,5Ø
         CNST=M(IXD)
         DO 17ØØ J=1,NV
          IF(J.EQ.IXD) GO TO 17ØØ
          CNST=CNST-(B(I,J)*S(IXD)/S(J))*M(J)
          X(J)=B(I,J)*S(IXD)/S(J)
17ØØ     CONTINUE
         X(IXD)=Ø.Ø
         IF(LOGF.NE.Ø) CNST=EXP(CNST)
         WRITE(9,56) FKMX(I),CNST,(X(J),J=1,NV)
56       FORMAT(1H ,F5.3,2X,G12.4,10(1X,E1Ø.3))
18ØØ     CONTINUE
         IF(LOGF.EQ.Ø) GO TO 2Ø5Ø
         WRITE(9,58)
58       FORMAT(1HØ/1HØ,"LOG-LINEAR MODEL: INTERCEPT CONVERTED TO ANTILOG")
C        GENERATE RIDGE TRACE
2Ø5Ø      DO 21ØØ I=1,5Ø
           DO 21ØØ J=1,51
            PLT(I,J)=1H
```

```
2100  CONTINUE
C     FIND MIN AND MAX NORMALIZED COEFFICIENT VALUES
      SM=+1E99
      BG=-1E99
      DO 2200 I=1,50
       DO 2200 J=1,NV
        IF(J.EQ.IXD) GO TO 2200
        IF(B(I,J).LT.SM) SM=B(I,J)
        IF(B(I,J).GT.BG) BG=B(I,J)
2200  CONTINUE
C     LOAD PLOT BUFFER
      XI=(BG-SM)/50.0
      DO 2400 I=1,50
       J1=1.0-SM/XI
       IF(J1.GT.0.AND.J1.LE.51) PLT(I,J1)=1H.
       DO 2400 J=1,NV
        IF(J.EQ.IXD) GO TO 2400
        J1=1.0+(B(I,J)-SM)/XI
        PLT(I,J1)=ALNUM(J)
2400  CONTINUE
C     PRINT RIDGE TRACE
      WRITE(9,60)SM,BG
60    FORMAT(1H1,"RIDGE TRACE: NORMALIZED COEFFICIENTS"/
     1  1H0,"COEFFICIENT RANGE:",F12.4," TO",F12.4/
     2  1H0,"K-VALUE",1X,51(1H.)," R-SQUARE"/)
      DO 2500 I=1,50
       WRITE(9,65) FKMX(I),(PLT(I,J),J=1,51),PLT(I,52)
65    FORMAT(1H ,F5.3,3X,51A1,F7.4)
2500  CONTINUE
C     OUTPUT VARIANCE INFLATION FACTORS (VIF)
      WRITE(9,70)(NN,NN=1,NV)
70    FORMAT(1H1,"VARIANCE INFLATION FACTORS FOR REGRESSION",
     1  " COEFFICIENTS"/ 1H0,"  VARIABLE:",16I7)
      WRITE(9,52)
      DO 2600 I=1,50
       WRITE(9,75)FKMX(I),(VIF(I,J),J=1,NV)
75    FORMAT(1H ,F5.3,6X,16F7.1)
2600  CONTINUE
      WRITE(9,80)
80    FORMAT(1H1)
      STOP
      END
```

VITA

Gregory Lagavero was born on 25 May 1954 in Buffalo, New York. He graduated from high school in Forestville, New York in 1972 and attended the University of Rochester from which he received the bachelor of arts degree in Statistics and Economics in 1977. After graduation, he was employed by the United States Air Force at Newark Air Force Station, Ohio. He entered the School of Engineering at the Air Force Institute of Technology in June 1981.

Permanent Address: 61 Indianwood Drive
Thornville, Ohio 43076


END